

ABSTRACT

Title of dissertation: MODELING AND ANALYSIS OF
 MASSIVE SOCIAL NETWORKS

Nan Wang, Doctor of Philosophy, 2005

Dissertation directed by: Professor Aravind Srinivasan
 Department of Computer Science

Traditional epidemiological research has focused on rate-based differential-equation models with *completely mixing* populations [6, 7, 44]. Although successful in explaining certain phenomena of disease spreading, the traditional approach is unable to deal with disease spreading in realistic massive social networks, where most people only mix locally with few other people. We have develop an approach based on a combination of network theory and discrete-event simulations to study epidemics in *large urban areas*, which do not assume complete mixing populations. Our results include (1) detailed structural and temporal analyses of the social contact networks produced by TRANSIMS [10], a simulator for detailed transportation/traffic studies; (2) realistic simulation of contagious diseases (e.g., smallpox) on the social contact networks through EpiSims [32], a simulation-based analytical tool to study the spread of infectious diseases in an urban environment; (3) identifying a number of new measures that are significant for understanding epidemics and for developing new strategies in policy planning; (4) introduction of random graph models for theoretical analysis of the structural and algorithmic aspects of the social networks; and (5) combinatorial formulations and approximation algorithms for performing quarantine, vaccination and sensor placement, as aids to decision-making.

The social network that we have mostly dealt with is for the city of Portland, Oregon,

USA, developed as a part of the TRANSIMS/EpiSims project at the Los Alamos National Laboratory. The most expressive social contact network is a bipartite graph, representing *people* and *locations*; edges represent people visiting locations on a typical day. We also build random graph models to generate a family of social networks by taking as input some basic parameters of the Portland social network, and analyze social networks generated by these models.

MODELING AND ANALYSIS OF MASSIVE SOCIAL NETWORKS

by

Nan Wang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Professor Aravind Srinivasan, Chair
Professor Alexander Barg
Professor Lise Getoor
Professor Jonathan Katz
Dr. V.S. Anil Kumar

© Copyright by
Nan Wang
2005

This dissertation is dedicated to my parents.

ACKNOWLEDGMENTS

It has been a pleasure and a fortune to work with Aravind Srinivasan these past few years. I will be forever in his debt for Aravind's agreeing to supervise me four years ago, whose decision made the whole change of my career and life. Being an academic advisor, Aravind skillfully taught me how to conduct research, unwaveringly supported me to attend various conferences, and persistently helped me to improve my communication and presentation skills. He continually recommended to me interesting problems one after another, of which each turned out to be a fruitful research direction to pursue and some definitely have become cornerstones of my PhD research. Going beyond supervising me in research, Aravind has become a mentor in many aspects of my life. Our discussions on a variety of topics have been just as enlightening to me: humanity, globalization, balance of happiness and achievements ..., to name a few. Even though being a man of broad knowledge and vast experience himself, Aravind never hesitated to ask me about how to improve himself in areas that he is not content with himself, although he has already done far better than what I could do. The influence of his personality and intelligence will remain in my life forever.

One of Aravind's biggest favors to me was to introduce me to Madhav Marathe. Madhav was kind enough to take me as an intern into Los Alamos National Laboratory in the summer of 2002, which started the whole dance of this thesis. Madhav is such a person who possesses both dispositions of being a diligent researcher and a sophisticated manager. He was always enthusiastic at explaining ideas and prospects of the big projects

to me, and repeatedly provided insightful advice on my work. As a researcher, Madhav worked as hard as others and helped me overcome technical difficulties one after another. As a manager, he consistently maintained the high quality of the overall projects and taught me how to present the ideas clearly and efficiently to outsiders. Although having to coordinate every aspect of the big projects, Madhav was always kind to discuss with me about the current progress and trustfully let me pursue independently in my research. His favors and help to me were not limited to academic achievements, but were beneficial in many aspects of my life. This dissertation would not be possible without Madhav's timely appearance in my PhD years. I owe him in more ways than can be enumerated here.

I also felt lucky to have met Anil Kumar and Stephen Eubank at Los Alamos National Laboratory, without whose help I could not imagine where this dissertation would go. Anil worked with me side-by-side on many problems and Stephen supported us whenever we needed it. Whenever I had a question or an idea I would come to Anil, and he was always happy to discuss with me and help me find ways to tackle it. His contribution to the formation of this dissertation is enormous. I was delighted when he agreed to join my committee as a special member.

I met Jonathan Katz in the middle of my PhD program, whose deep understanding of cryptography and broad knowledge of theoretical computer science impressed me a lot. I am grateful for his showing me the wonderful world of cryptography. Jonathan always recommended the relevant papers for me to read and posed me challenging yet appropriate problems to work on. Under his guidance, I was able to coauthor with him and published a cryptographic paper in a top conference. Parallel to his intellectual achievement is his ability and willingness to teach and work with students and unselfishly share with them

his intelligence and knowledge (after making them think hard). I will forever remember the things I learned from Jonathan. I felt fortunate and honored to have been able to work with him.

Thanks are also due to my committee members, Professors Aravind Srinivasan (committee Chair), Alexander Barg (Dean's representative), Lise Getoor, Jonathan Katz, and Dr. V.S. Anil Kumar, for their serving on my committee and sparing their invaluable time reviewing the manuscript. I also thank David Mount for serving on my preliminary examination committee.

This research was supported in part by the National Science Foundation under Grant No. 0208005 and while visiting Los Alamos National Laboratory. I acknowledge these sources for their support.

I was very fortunate to have the chance of working with many dedicated fellow students and knowledgeable faculty members. Samir Khuller's advanced algorithms course was the first one I took at this department. I enjoyed a lot interacting with him. Howard Elman not only taught me technical stuff through his class, but also influenced me immensely by his great personality. I shared the office with Yung-Chun Justin Wan, Rajiv Gandhi, Yoo-Ah Kim, and later Srinivas Kashyap for several years. I appreciate the harmonious atmosphere we all created and maintained in this office. Special thanks to Rajiv Gandhi for his encouraging me to ask Aravind Srinivasan for supervision. Among those who shared the experience with me at Maryland and made my time here much more enjoyable, I want to specifically mention Bin Gan who always generously helped me whenever I needed it. Feng Guo, Qing Xie and Woei-Jyh Adam Lee also helped me a lot. Omer Horvitz always brought laughs to me and I also brought some back to him. Srinivasan Parthasarathy shared the intern experience with me at Los Alamos National

Laboratory in summer 2002, we had a lot of fun ever since. Shang-Chieh Wu broadened my knowledge of systems and C++. He also did me a big favor of taking pictures of me during the commencement day. In addition to all the above mentioned people, my time at the University of Maryland was made enjoyable also by the following folks: Fatima Bangura, Adam Bender, Indrajit Bhattacharya, Vasile Gaburici, Bill Gasarch, Vijay Gopalakrishnan, Jodie Gray, Gwen Kaye, Chiu-Yuen Koo, Gang Liang, Chunyuan Liao, Ruggero Morselli, Xue Wu, Jihwang Yeo, and Ping Yu.

Although six years have passed since I left Texas Tech University, my first stop in America, I still cannot help mentioning Wijesuriya Dayawansa for his guidance and support. I thank him for first seeing my potential and talent and encouraging me to pursue my dreams when I decided to apply for PhD programs elsewhere. I also want to mention James Dunyak who was my first advisor in America and financially supported me throughout most part of my time at Texas Tech. James was such an amenable person that he gave me much freedom to learn and do research according to my interests. Thanks are also due to Zhimin Zhang who encouraged and helped me a lot for pursuing further study after my Master's program at Texas Tech. All three of them caused my dream to happen and made my subsequent experiences at Maryland possible.

I spent one summer at IBM T. J. Watson Research Center, one summer at Fujitsu Labs of America, and two summers at Los Alamos National Laboratory as an intern. These experiences have been invaluable in building up my career path and have played an important role in my job hunting in the last semester of my PhD program. I enjoyed a lot of all my intern experiences and I thank my mentors and colleagues there who made it possible: Mark Wegman, V.T. Rajan, Doug Kimelman, Ching-Fong Su, Madhav Marathe, Anil Kumar, Stephen Eubank, Chris Barrett, and James Smith. I also thank the

department of Computer Science for financial support through the teaching assistantship.

Finally, thanks to everyone who shared this journey with me.

TABLE OF CONTENTS

List of Tables	x
List of Figures	xi
1 Introduction	1
1.1 Preliminaries	5
1.2 Organization and Contributions of this Thesis	8
2 Related Work	15
2.1 Traditional Mathematical Epidemiology	15
2.2 Complex Networks and Random Graph Models	16
2.3 Dynamics and Algorithms on Social Networks	19
3 TRANSIMS and EpiSims	22
3.1 TRANSIMS	22
3.2 EpiSims	27
4 Structures of Portland Graph and Strategies to Minimize Disease Spreading	30
4.1 Basic Structure	31
4.2 Graph Expansion and Shattering	35
4.3 Overlap Ratios and the Sensor Placement Problem	37
4.4 Other Structural Properties	40
5 Random Graphs Models For Social Networks	46
5.1 Chung-Lu's Model	47
5.1.1 FASTGREEDY in Chung-Lu's Model	47
5.1.2 Domination by Sampling	52
5.2 Configuration Model	53
5.2.1 FASTGREEDY in the Configuration Model	58
5.3 FASTGEN Model	59
5.3.1 Analytical Validation of FASTGEN-1	63
5.4 Empirical Comparison of Generated Graphs and Portland Data	66
5.4.1 Degree Distributions	68
5.4.2 Overlap Ratios and the FASTGREEDY Algorithm	70
5.4.3 Shortest Paths and Clustering Coefficients	73
5.5 A Generic Framework	74
6 Algorithms for Quarantining and Vaccination Problems	78
6.1 Modeling the Efficacy of Vaccination Policies	78
6.2 The Quarantining and Vaccination Problems	79
6.3 Bicriteria Results for QP	83
6.4 A Bicriteria approximation for VP.	84
7 Simulation Based Dynamic Analysis	87
8 Conclusions and Future Work	91
8.1 Future Work	92

LIST OF TABLES

4.1	Means and standard deviations of fractions of shortest paths of different lengths in the shortest-path spanning trees sampled from the giant component of Portland's G_P graph.	34
5.1	Comparing the basic structures for the Portland data and randomly generated graphs: number of locations, size of the giant connected component of locations, size of the giant connected component of people, number of edges in the bipartite graph, average degree of people in the bipartite graph, time of generating a graph. The percentages are the percents of the quantities of generated graphs compared to those of Portland data.	68
5.2	Performance of FASTGREEDY and GREEDY (see 1.1.5) for the Portland data and graphs generated by Chung-Lu's model and the FASTGEN models. Percentages denote the sizes of the dominating sets (compared to the whole set of locations). Seconds and hours denote the time needed to run these algorithms.	72
5.3	Means and standard deviations of fractions of shortest paths of different lengths in the single-source shortest-path spanning trees sampled from the giant component of the people-people graphs.	74
5.4	Means and standard deviations of clustering coefficients of sampled vertices in the giant components of people-people graphs. Lower bounds and approximated lower bounds to the clustering coefficients of giant components of people-people graphs.	74
5.5	Numbers of edges, triangles, and length-2 paths, and clustering coefficients in the people-people graph (1615860 people) in the Portland data and the Chung-Lu's model.	74

LIST OF FIGURES

3.1	Figure showing, for a randomly chosen synthetic individual (whose person-id is pid=838206 as shown in the figure) in Portland, the activities, their locations (marked with lid's as their location-id's) and their time durations (in hour) over the course of 1.5 days. The x -axis indicates the time when an activity happens, the y -axis indicates the number of people in the same location where the activity is taken by the chosen individual. In Portland data there are nine activity types, namely home , work , shop , visit , social , other , serve , school , college . Each edge in the people-location bipartite graph is labeled by one of these activities.	24
3.2	An illustration of various steps in TRANSIMS, and the networks that are constructed. The squares denote locations, with the letters specifying the type of locations (H - home, W - work, C - carpool, L - place for lunch). The circles denote people, moving from one location to another. The thin solid lines between people and locations show the edges between people and locations, i.e., the edges of the graph G_{PL} – these edges are labeled with the time duration when the person is present at the location. The thick solid lines (with the arrows) show the “trajectory” of a person (four persons in this figure), as they move from one location to another; this also illustrates the temporal aspects of the network – for instance, person-1 moves from home to a carpool (viewed as a location), to a work location, and so on. At a location, all people who are at that location at the same time are connected by dashed edges - these are the edges of the people-people contact graph. The people also have labels – one such label (40 years old, male) is shown in the figure.	25
4.1	Degree distributions of locations and people in the bipartite graph G_{PL} for Portland data. The location degrees range from 1 to 7091, people degrees range from 1 to 15.	31
4.2	Degree distribution in the people-people graph for the Portland data.	32
4.3	Expansion of the people-people graph: the plots marked “Vertex expansion-2” and “Edge expansion-2” show the vertex and edge expansion for the graph G_P , while “Vertex expansion-1” and “Edge expansion-1” show the corresponding quantities in the graph obtained by retaining only those edges that involve an interaction of at least 1 hour. This leads to a much sparser graph and correspondingly lower values of vertex and edge expansions. . . .	36
4.4	Size of the giant connected component after each iteration.	37
4.5	Overlap ratios and performance of the fast greedy algorithms for the dominating set problem (Portland data).	41

4.6	Temporal degree distributions for different types of activity locations. The types of activity locations are: home , work , school , shop , social , college . Each plot contains the temporal degree of four randomly chosen activity locations of a given type, where the x -axis shows the time in hours, and the y -axis shows the number of people at that time (x) and that location (doing the specific activity). For example, the top left plot shows the temporal degree distribution for four randomly chosen home locations. The home location is really a block of homes, and that accounts for the large sizes.	42
4.7	Distribution of activity lengths for work , shop , social , school activities. For each activity type, and for each possible duration of times d , the plot shows the fraction of this activity lasted for d time units.	43
4.8	Contacts with each age-group, for people of ages 16, 30, 40, and 60. For each of these age groups (say A), and for each possible age-group (say B) on the X -axis, the plot shows the average number of contacts that group A makes with group B in the Y -axis. The average for a given age is computed by computing a distribution for each person p of that age and then summing up these distributions and dividing the resulting values by the total number of individuals of that age.	45
5.1	Description of the configuration model for generating random graphs.	53
5.2	Description of the algorithm for fast generation of random graphs.	62
5.3	Comparison of (bipartite) degree distributions in the Portland data and graphs generated by Chung-Lu's model, FastGen-1, and FastGen-2.	69
5.4	Comparison of degree distributions of people-people graphs of Chung-Lu's model, Portland data, the configuration model, FASTGEN-1, and FASTGEN-2.	70
5.5	Overlap ratios of locations in the Portland data and the four models.	71
5.6	Performance of FASTGREEDY for the dominating set problems.	72
6.1	The transformations done in the reduction for VP.	81
6.2	A Bicriteria Algorithm for QP	83
6.3	A Bicriteria Algorithm for VP	84
7.1	Cumulative number of deaths per number of initial infected, in case of a smallpox outbreak in downtown Portland, under a number of different response strategies.	90

Chapter 1

Introduction

The rapidly expanding population of today's towns and cities has resulted in very high population densities and social connectivity. Understanding the urban social-contact structure is critical for social scientists, urban planners, infrastructure companies, and governments. For example, the spatial distribution of population in a city, people's movement-patterns and their phone-calling patterns have a direct bearing on how wire-line and wireless infrastructure providers design their networks. Similarly, the social contact network also determines the spread of an infectious disease. For instance, one of the astonishing features of the recent SARS outbreak was the speed and the extent with which the epidemic spread: this is a demonstration of how "connected" the world has become. For many societies, this raised the important questions of how to detect the disease quickly, and how to control it (by quarantining, vaccination etc.). Because of the significant logistics involved, this needs to be done with as little cost as possible. Questions of this sort translate to classical ones of domination and cuts, which can be solved effectively only with a good knowledge of the structure of the underlying networks. Infrastructure planners, e.g., people designing transportation or communication networks, are also faced with questions that require an understanding of the structure of social networks. This thesis proposes an analytical and empirical study of various structural, temporal, and algorithmic issues of massive social networks.

A large part of this work is related to the EpiSims project. EpiSims is a simulation-based analytical tool developed at the Los Alamos National Laboratory (LANL) [32] to

study the spread of infectious diseases in an urban environment. In order to understand the spread of contagious diseases, we need a realistic representation of a social network. The TRansportation ANalysis and SIMulation System (TRANSIMS) developed at LANL produces estimates of the social network in a large urban area based on the assumption that the transportation infrastructure constrains people’s choices about what activities to perform and where to perform them [10]. TRANSIMS combines varied data sets such as census and land use data, activity surveys, etc. and produces locations of all travelers on a second-by-second basis; see [10] for more information. EpiSims is a tool for simulating the spread of epidemics at the level of individuals in a large urban region, taking into account realistic contact patterns and disease transmission characteristics. It integrates models for disease propagation within a host, transmission between hosts, and contact patterns among hosts to create a realistic microscopic epidemic simulation. It provides estimates of both the geographic and demographic distribution of disease as a function of time. EpiSims provides a natural representation for assessing the capability of public health infrastructure to defend urban populations against disease outbreaks. Tools such as vaccination campaigns, quarantine and isolation, and contact tracing can be modeled easily within the simulation, even if they are demographically targeted (e.g., immunizing children). For example, EpiSims can be used to assess the impact on the population at large of immunizing children against influenza, or the logistical support required to ensure that a ring-vaccination strategy can contain a smallpox outbreak. (Ring-vaccination is a simple and natural strategy of enclosing an area of disease-outbreak by a “ring”, and vaccinating everyone on the ring.)

Structural analysis of social contact networks is an integral part of EpiSims and TRANSIMS. There are three key parameters to develop: (1) new graph properties that

serve as structural invariants and model a given policy/design question at hand, (2) mathematical theories that explain the evolution of these graphs and also serve as models generating random ensembles of such networks, and (3) fast computational methods for solving problems arising in (1) and (2). For example, recent results have shown that many real-world social and infrastructure networks cannot be modeled by the Erdős-Rényi random graph model [28]. New mathematical models have been proposed for generating generic “real world networks”. This has given impetus to research on the structural properties of social networks, and theories of their evolution. Since exact data is hard to obtain, much of the research has focused on constructing random graph models that match available data for measures like degree distributions, clustering coefficients, etc. It is also likely that there would be no universal model. As mentioned above, the Erdős-Rényi random graph model [28], which has been found to be very useful in many contexts, is unrealistic for social networks. Most social networks have been observed to have power law degree distributions (see Section 1.1 for definitions), while Erdős-Rényi model has a flat degree-distribution with exponential tails.

Recent work on structural analysis of large real-world networks started with experimental work on the structure of the Internet/World Wide Web (see, e.g., [2, 17, 34]). One of the striking results of such studies was the power-law nature of the degree distribution. Since models such as Erdős-Rényi model do not have this property, these studies have sparked research on theoretical models to explain such properties. These models are broadly of two types: (1) incremental models for constructing such graphs, using primitives such as preferential attachment (see, e.g., [8, 16, 23, 50, 51]) wherein each *new* vertex chooses its set of neighbors (*old* vertices) randomly according to some function of the degrees of old vertices; and (2) models which assume a given degree sequence, and generate

random graphs either by placing each edge independently, with a probability dependent on the degrees of the end-points [21], or generating a graph uniformly at random from the space of graphs with the given degree-sequence [56].

Classical structure properties of networks include: degree distribution, clustering coefficient distributions, shortest-path distribution, and connected components. Most of the work on the above models has been toward understanding such structural properties [16, 21, 23, 56].

In addition to such structural results, there has been much work in classical random graph theory on algorithms for problems that are NP-hard in the worst case [36]. However few studies seem to have been carried out for random graph models for social networks. Combinatorial problems such as dominating sets, cuts and flows are useful in this setting. One challenge here is that these graphs are typically very large (of the order of a million nodes or more), since they represent models for social interaction, and computing even simple quantities on such massive graphs is non-trivial. This motivates the need for very fast but still accurate algorithms. Another important problem that has not received attention in this context so far is that of efficient generation of these graphs. For instance, in the model of [21] (the *Chung-Lu model*), straight forward implementations require quadratic running time in terms of the number of nodes, even when the average degree is small. For very large graphs with millions of nodes, a fast-generation mechanism that takes time linear in the number of edges, but still produces graphs with meaningful structure, would be very useful (e.g., for sensitivity analysis where multiple runs are needed).

1.1 Preliminaries

Since our work involves probabilistic methods, random graph models, approximation and randomized algorithms, we add some preliminaries in this section before using them in the upcoming sections.

The starting point of our work is a realistic social contact structure based on data from Portland, Oregon, USA. The main component of this dataset that is relevant to us is a social contact network $G(P \cup L, E)$, which is a bipartite graph with a set P of people and a set L of locations.¹ An edge (p, ℓ) is present if person p visits location ℓ on a typical day. The dataset is massive: $|P| \sim 1.6$ million; $|L| \sim 1.8 \times 10^5$, and $|E| \sim 6$ million.

Definition 1.1.1 (Bipartite graph) *A bipartite graph $G(P \cup L, E)$ is a graph whose vertices can be partitioned into two parts, P and L , such that the set of edges are $E \subseteq \{(p, \ell) : p \in P, \ell \in L\}$. In other words, edges only exist between P and L .*

Definition 1.1.2 (Approximation algorithm) *Let $OPT(I)$ denote the cost of an optimum solution to an instance I of an optimization problem \mathcal{O} . We say that a polynomial-time algorithm \mathcal{A} is a ρ -approximation algorithm for the optimization problem \mathcal{O} if and only if for every instance $I \in \mathcal{O}$, letting $\mathcal{A}(I)$ denote the cost of the solution returned by \mathcal{A} on I ,*

- $\mathcal{A}(I) \leq \rho \cdot OPT(I)$, $\rho > 1$, when \mathcal{O} is a minimization problem;
- $\mathcal{A}(I) \geq \rho \cdot OPT(I)$, $0 < \rho < 1$, when \mathcal{O} is a maximization problem.

An example of an NP-hard problem that has an $O(\log(n))$ -approximation algorithm is the dominating set problem [71], where n is the number of sets. Our work on placing

¹For diseases such as smallpox L can be restricted to buildings; the contact times in city public transport are usually inadequate for spread of the disease.

sensors to detect disease (see Section 4.3) is equivalent to the dominating set problem. We present this problem in the following form that fits our context and describe a classical greedy algorithm that has an $O(\log(n))$ -approximation ratio.

Problem 1.1.3 (Dominating set problem) *For a bipartite graph $G(P \cup L, E)$, a dominating set $L' \subseteq L$ is a subset of L such that every $p \in P$ is adjacent to some $\ell \in L'$; p is said to be dominated by L' . The dominating set problem is to find such a minimum sized set $L' \subseteq L$ that dominates the whole set P .*

Definition 1.1.4 *For a vertex v of a graph $G(V, E)$, let $N(v)$ be the set of vertices adjacent to v , i.e., $N(v) = \{u : (v, u) \in E\}$.*

The following algorithm, GREEDY, is a classical greedy algorithm for the dominating set problem [71].

Algorithm 1.1.5 (The GREEDY algorithm) *Repeatedly do the following until all people P have been dominated: Find a maximum-degree location $\ell \in L$; include ℓ in the dominating set L' ; set $L \leftarrow L \setminus \{\ell\}$ and $P \leftarrow P \setminus N(\ell)$, where $N(\ell)$ is the set of people in P adjacent to ℓ .*

The running time of the above algorithm is $O(|P| \cdot |L|)$. Algorithm 1.1.5 is an $O(\log |L|)$ -approximation algorithm to the dominating set problem. The proof can be found in [71]. In general, the $\log |L|$ factor cannot be improved [35].

In practice, it is sufficient to relax the requirement of dominating *all* people to dominating *a large fraction* of people to obtain more efficient strategies. Below is a version reflecting this trade-off.

Problem 1.1.6 ((1 - ϵ)-dominating set problem) For a bipartite graph $G(P \cup L, E)$ and a given parameter $0 < \epsilon < 1$, find a minimum set $L' \subseteq L$ that dominates at least a $(1 - \epsilon)$ fraction of people P .

Most of our arguments are probabilistic. The Chernoff-Hoeffding bound [19, 45] plays an important role in our reasoning. The following lemma is from page 70 of [58].

Lemma 1.1.7 (Chernoff-Hoeffding bound) Let X_1, X_2, \dots, X_n be independent 0-1 random variables such that, for $1 \leq i \leq n$, $\Pr[X_i = 1] = p_i$, where $0 < p_i < 1$. Then, for $X = \sum_{i=1}^n X_i$, $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$, and $0 < \delta < 1$,

$$\Pr[X < (1 - \delta)\mu] < e^{-\mu\delta^2/2}.$$

For simplicity of notation, we write this as “Chernoff bound” instead of “Chernoff-Hoeffding bound”.

In this thesis, lots of empirical and analytical results are related to power law distributions in social networks.

Definition 1.1.8 (Power law distribution) Let $\mathcal{P} = \{p_k\}_{k=1,2,\dots,n}$ be a discrete probability distribution, where $\Pr[X = i] = p_i$. Given a parameter $\alpha > 0$, let $\zeta(\alpha; n) = \sum_{k=1}^n \frac{1}{k^\alpha}$. We say that \mathcal{P} is a power law distribution with parameter α if $p_k = \frac{k^{-\alpha}}{\zeta(\alpha; n)}$ for all $k = 1, 2, \dots, n$.

A popular parameter for measuring in a social network is *clustering coefficient* [72] or *transitivity*. It measures how dense a network is *locally* (with respect to the neighborhood of a node), for example, how likely a person’s friends are also friends among themselves.

Definition 1.1.9 (Clustering coefficient) In a graph $G(V, E)$, the clustering coefficient $C(v)$ for a node v of degree at least two is the ratio of the number of edges $(u, w) \in E$,

where $u, w \in N(v)$, and the maximum possible number of edges in $N(v)$, i.e., $\frac{d(v)(d(v)-1)}{2}$, where $d(v) = |N(v)|$ is the degree of v . The clustering coefficient of the whole graph is the average of the clustering coefficients of all nodes, i.e., $C(G) = \sum_{v \in V} C(v) / |V|$.

Another definition of clustering coefficients taking the graph as a whole is as follows.

Definition 1.1.10 (Global clustering coefficient) Let $\Delta(G)$ be the number of triangles in graph $G(V, E)$, and let $\Gamma_2(G)$ be the number of length-2 paths in G (i.e., the number of connected triples of vertices in G). The global clustering coefficient $GC(G)$ is

$$GC(G) = \frac{3 \times \Delta(G)}{\Gamma_2(G)}. \quad (1.1)$$

We use random graph models to analyze the empirical data and results in a rigorous way. Although the classical Erdős-Rényi model cannot capture the basic property of social networks, i.e., power law distributions of degrees, its idea of modeling extends to the random graph models that are of interest to us. The description below is from [5].

Definition 1.1.11 (Erdős-Rényi model) Let n be a positive integer, $0 \leq p \leq 1$. The random graph $G(n, p)$ is a probability space over the set of graphs on the vertex set $\{1, \dots, n\}$ determined by

$$\Pr[(i, j) \in G] = p$$

with these events mutually independent. Note that the pairs (i, j) are undirected edges.

1.2 Organization and Contributions of this Thesis

This thesis is organized into eight chapters. We survey the work related to the research presented in this thesis in Chapter 2.

Since our work is based on the two simulators TRANSIMS [10] and EpiSims [32], we describe about them in Chapter 3. In Section 4.1 of Chapter 4, we describe some classical structural measures of a social network, and sampling methods for computing these measures. These classical measures include ones studied by social network and complex network communities, i.e., degree distributions, connectivity, shortest paths, and clustering coefficients. Besides that, we also study some new properties like vertex and edge expansion (Section 4.2), and structures related to temporal and demographic properties of the social network (Section 4.4). It is observed in this social network that the degree distribution of locations resembles the power law distribution with parameter $\alpha \approx 2.8$ (see Definition 1.1.8) which is between *two* and *three* as observed in many other social networks (e.g., the Internet graph), while the degree distribution of people resembles a Poisson distribution with a small mean *four* and upper-bounded also by a small value *fifteen*. This social network also has a small-world structure [55, 69, 73, 74], where most shortest paths have lengths at most *four*. Like many other social networks, the clustering coefficient (see Definitions 1.1.9 and 1.1.10) of the Portland data is quite high (around 0.45). Although measures like shortest path distributions and clustering coefficients are achievable by simple polynomial time algorithms, given the large-scale size of the social network, their quadratic running time is not practical at all. Furthermore, there are generally no polynomial time algorithms to compute exactly the vertex and edge expansions. Instead of measuring them exactly, we present efficient sampling methods to estimate these structures accurately with high probability [30]. These new structures and others studied in this thesis (e.g., the overlap ratios) give us a better understanding of realistic and massive social networks and shed light on new and practical strategies against outbreaks of infectious diseases in large urban areas. Most work in this chapter was done in

[30, 31].

After a preliminary exploration of the structural properties of the Portland social network, we turn our attention to the main purpose of this thesis – designing strategies for preventing outbreaks of infectious diseases in large urban areas. In Section 4.2 we show that a simple vaccination and quarantine strategy is not practical in real life. We then present an efficient disease detection strategy of placing sensors at *selected* locations. We also show that mass vaccination might not be needed in the presence of a better disease detection strategy. The high expansion rate in Section 4.2 explains why mass vaccination might be unavoidable, as observed in our simulation of various vaccination and quarantining strategies. Some diseases can be detected by installing sensors at locations to monitor people in them. This leads to the following natural problem: what is the smallest set of locations where sensors can be placed to detect any infected person. This, in turn, is equivalent to the *dominating set problem* (see Problem 1.1.3). This problem is generally *NP-hard* [39] and cannot be approximated to within $(1 - \Omega(1)) \ln |L|$ unless $P=NP$ [35], where $|L|$ is the number of locations. By introducing a new measure, *the overlap ratio* [30] (see Section 4.3), we find that in the Portland data the sets of people visiting different locations don't overlap much, which is much different from the worst case for the dominating set problem. This special structure favors those greedy algorithms that choose large locations into the dominating set. We simulate the traditional greedy algorithm (see Algorithm 1.1.5) for the dominating set problem on the Portland data and obtain desirable results: with few locations being installed with sensors, a large fraction of people are dominated. By the virtue of overlap ratios of Portland data, we design fast greedy algorithms running in nearly linear time (see Section 4.3) that perform almost as well as the traditional one [30] but run much faster than it.

All the above are empirical works focused on a particular data set. In order to generalize our results to a large class of realistic urban social networks and provide theoretical models to explain the formation of these networks as well as rigorously analyze strategies against disease spreading in them, we present in Chapter 5 two random graph models, *Chung-Lu's model* [21] and *the configuration model* [56], that generate random networks resembling the Portland data. Both models take as input the two degree sequences of people and locations of the Portland data and *randomly* generate graphs that (approximately) match the given degrees. Chung-Lu's model creates an edge between a person p and a location ℓ with probability $\frac{d(p)d(\ell)}{\sigma}$, independently of other pairs of people and locations, where $d(p)$ and $d(\ell)$ are the given degrees in the input degree sequences, and $\sigma = \sum_{p \in P} d(p) = \sum_{\ell \in L} d(\ell)$ is the number of edges specified by the degree sequences of the bipartite graph. Chung-Lu's model generates random graphs whose *expected* degrees match the given degrees. Despite its virtue of describing the random process of the formation of social networks, Chung-Lu's model suffers from poor scalability of generating graphs. The running time of generating a graph is in the order of $|P| \cdot |L|$ where $|P|$ is the number of people and $|L|$ is the number of locations. We present a different and much faster (in terms of generation time) model whose generation time is at most $O(\sigma \cdot \log |P|)$ and preserves most of Chung-Lu's model's properties, except the independence of creating edges. Instead, the edges in our model are negative correlated. Our model is a careful implementation of the approach of [68]. Despite this difference, all the arguments we make in Chung-Lu's model carry on to our fast generation model, via an extended Chernoff bound [67].

The configuration model [56] is different from Chung-Lu's model in both of the generation process and the degree matching. Unlike Chung-Lu's model where each pair

of people and locations is checked for creating an edge or not, the configuration model randomly wires the input edges but still preserves the degrees of people and locations.² The generated graphs match the input degrees *exactly*, and each *simple* graph of the given degree sequence has equal probability of being generated (see 5.2.1). The generation time is of the order of the number of edges, which is much faster than Chung-Lu’s model. Despite its fast generation time, exact matching of degrees, and uniform distribution of generated (simple) graphs, the configuration model suffers from multi-edge occurrence in the random process. On the other hand, the probability of the occurrence of edges between a person and a location can be calculated easily and is *approximately* the same as that in Chung-Lu’s model, i.e., $\frac{d(p)d(\ell)}{\sigma}$.

Besides matching the given degree sequences, we also show that many other important structures (which are not specified in the input) are matched closely between graphs generated by these models and the Portland data. These structures include connectivity, shortest paths, clustering coefficients, vertex and edge expansions, overlap ratios, and the performance of the fast-greedy and traditional greedy for the dominating set problem. These matchings confirm the validity of the models from an empirical perspective.

Despite the close approximation of probabilities of occurrences of edges between people and locations, the starting points and random processes of generating them are quite different between these two models. We argue in Chapter 5 that Chung-Lu’s model may capture the formation process of social networks, which seems harder with the configuration model. The probabilities of occurrences of edges in many social networks, however, may not be the same as those specified in Chung-Lu’s model. In order to generate ran-

²Actually, multi-edges may occur in this model and will be counted into the degrees. If multi-edges are not allowed, one has to repeatedly run the generation process until a simple graph is generated.

dom graphs with any arbitrary given probabilities of occurrences of edges, and still match the given degrees (at least) in the expectation values, we present in Chapter 5 a generic framework that uses the approach of [37] as a building block. Since the running time of the approach of [37] can be as large as cubic, it is not practical at all, we present a *decomposition* approach that divides the original big social networks into many smaller ones and generates each one *independently*. This approach is related to the edge partitioning problem, and can be solve efficiently via approximation algorithms in [38, 42]. The independence among the subgraphs implies that the generating process can be parallelized. With a good decomposition, this approach can significantly shorten the running time, even without parallelization. Furthermore, the decomposition also captures the division and community property of social networks. When the input probabilities of occurrences of edges is the same as those in Chung-Lu’s or the configuration models, these two models can also serve as the building blocks.

Armed with these models, we present, also in Chapter 5, a rigorous analysis of the empirical work done in the previous chapters, including the performance of our fast-greedy algorithm for choosing locations to place sensors in. We prove that for both Chung-Lu’s model and the configuration model, with specific given degrees resembling those in the Portland data, i.e., power law distribution (with bounded values) for locations and Possison distribution (upper bounded by a constant) for people, our fast-greedy is a $(1+\epsilon)$ -approximation algorithm to the optimum solution, with high probability (dependent on ϵ), where $0 < \epsilon < 1$ is a small value.

After both empirical evaluation and rigorous analysis of the fast-greedy algorithm for the sensor placement problem, we come back to the vaccination and the quarantine problems in social contact networks. In Chapter 6 we show that both of these two problems

are *NP-complete* and give a bicriteria approximation for them using network flows [71].

Our results on high expansions (see Section 4.2) suggest that the disease is likely to spread quickly if it is not controlled at an early stage. However, exactly how the number of casualties depends on response delay and what constitutes *early enough* depend on disease-specific factors such as incubation period and probability of transmission, as well as scenario-specific factors such as the means of introduction. Because these dependencies cannot be easily determined from analysis of the static social network (see Chapters 4–6), we fall back to the two simulators, TRANSIMS and EpiSims, described in Chapter 3. In Chapter 7 we use EpiSims [32] to simulate Smallpox spreading on the social network (Portland data) generated by TRANSIMS [10]. The study shows that time of withdrawal to the home is by far the most important factor, followed by delay in response. This indicates that targeted vaccination (see Section 4.2 and Chapter 6) is feasible when combined with fast detection (see Section 4.3 and Chapter 5). Ironically, the actual strategy used is much less important than either of these factors. Overall, these results suggest a much greater efficacy for targeted strategies than suggested by the results of Kaplan, Craft, and Wein [46].

We conclude and describe future work in Chapter 8.

Chapter 2

Related Work

2.1 Traditional Mathematical Epidemiology

The explosive growth of urban population in the past century has led to a variety of new problems related to public health: the high density of people and their interaction lead to a significant risk of epidemics [76]. This was evident in the recent SARS epidemic, and is a testimony to the “small world” nature of today’s society [55, 70]. Coupled with recent fears of bio-terror attacks, there has been a spurt of research on understanding epidemics and techniques to aid policy planning, e.g., for vaccination, quarantining and disease-detection strategies.

Traditional epidemiological research has focused on rate-based differential equation models on completely mixing populations in which all the people are allowed to interact with each other. For example, the SIR model of epidemic disease [6, 7, 44] divides the population into three classes [60]: susceptible (S), meaning they don’t have the disease of interest but can catch it if exposed to someone who does; infected (I), meaning they have the disease and can pass it on; and recovered (R), meaning they have recovered from the disease and have permanent immunity, so that they can never get it again or pass it on. In traditional mathematical epidemiology [6, 7, 44], one then assumes that any susceptible individual has a uniform probability β per unit time of catching the disease from any infective one and that infective individuals recover and become immune at some stochastically constant rate γ . The fractions s , i and r of individuals in the states S, I,

and R are then governed by the differential equations

$$\frac{ds}{dt} = -\beta is, \quad \frac{di}{dt} = \beta is - \gamma i, \quad \frac{dr}{dt} = \gamma i. \quad (2.1)$$

An attractive feature of this modeling approach is that it allows one to obtain analytical expressions for a number of interesting parameters such as the numbers of sick, infected and recovered individuals in a population. It also illustrates some basic dynamics of diseases. Such a modeling approach, however, does not capture the complexity of human interactions that serve as a mechanism for disease transmission, because in reality diseases can only spread between those individuals who have actual physical contact, and the structure of the contact network is important to the pattern of development of the disease [60]. In addition, typically the number of different sub-population types considered is small (for analytical tractability) and parameters such as mixing rate and reproduction number ¹ are either unknown or hard to observe.

2.2 Complex Networks and Random Graph Models

In this thesis, we outline an approach based on a combination of network theory and discrete event simulations to study epidemics in *large urban areas*. The main idea is that a better understanding of the characteristics of the social contact network can give better insights into disease dynamics and vaccination/quarantining strategies, which can be used in the epidemic simulation. For instance, our recent work in [29] shows that a better understanding of the underlying network structure leads to more refined conclusions, e.g.

¹The threshold for many epidemiology models is the basic reproduction number R_0 , which is defined as the average number of secondary infections produced when one infected individual is introduced into a host population where everyone is susceptible [44, 24]. For many deterministic epidemiology models, like the SIR model, an infection can get started in a fully susceptible population if and only if $R_0 > 1$.

in some cases mass vaccination might not be needed in the presence of a better disease detection strategy. Similar work by Meyers *et al.* [54] has demonstrated that new insights on the disease dynamics can be obtained by understanding the contact structure carefully. Interestingly, the first reported analysis of social networks in urban regions for effectively containing the spread of Cholera was done in the late 1800s in London [12]: in this outbreak, a map was used to chart the outbreak and relate it to a contaminated water pump; shutting down the pump immediately brought the outbreak to an end. The earliest formal work on use of network structure for epidemiological studies appears to be that of Elveback, Fox and Ackerman [27]. Recently, a number of other authors have also undertaken a similar approach [53, 65]. In contrast to the work of [53, 54], we study *realistic and large* urban social contact networks for an entire city consisting of well over a million people.

Recently there has been a resurgence of research in complex networks: the renewed interest is driven by a number of empirical and theoretical studies showing that network structure plays a crucial role in understanding the overall behavior of complex systems. See [1, 3, 8, 17, 54, 60] and the references therein for recent results in this active area. However, properties of social contact networks that are crucial for understanding epidemics have been explored only recently [59, 60, 61, 62, 54]. Another recent direction of research has been to determine random graph models that can generate such networks: the traditional Erdős-Rényi model [28] (see Definition 1.1.11) does not capture many important features of real networks, such as power law degree distributions, high clustering coefficients, etc. Unfortunately, as we argue in the following, many of these random graph models, such as the preferential attachment model [8], are not suited for social network analysis either.

The first highly influential model trying to explain the formation of power law degree

distribution is the preferential attachment model [8]. This is an evolutionary (or growth) model, where nodes are added sequentially. Each node chooses some existing nodes to connect to via edges. These nodes are not chosen uniformly, but with a higher preference for nodes that already have a higher degree. These models have been shown to be very useful for modeling the Internet web graph, but are not very promising for social networks for the following two reasons. The first is that social networks do not have very strict power laws. For example, in the Portland data, only the distribution of location degrees in the bipartite graph follow a power law in some range, but people degrees (in the bipartite and the projected contact graphs) don't follow a power law. Furthermore the power law parameters (see Definition 1.1.8) of degree distributions generated by preferential attachment models are not adjustable – they cannot be predefined as input to the models. The second reason is that preferential attachment models are not defined for bipartite graphs, and it is not clear whether they can be modified for bipartite graphs. Generating instances of graphs is also very slow (quadratic running time) in this kind of evolutionary model. Nonetheless, the model of Barabási and Albert [8] has attracted an exceptional amount of attention in the literature. In addition to analytic and numerical studies of the model itself, many authors have suggested extensions or modifications of the model that alter its behavior or make it a more realistic representation of processes taking place in real-world networks. Newman [60] gives a good survey of those works.

Instead of trying to explain the formation of graphs with a certain degree distributions, another line of research in modeling social networks is to generate random graphs *matching* the given degree sequences. There are two representative models along this line. The first one is the configuration model. It was first introduced by Bender and Canfield [13], refined by Bollobás [15] and also Wormald [75]. It randomly places edges but

still preserves the degrees of people and locations. The generated graphs match the input degrees *exactly*, and each *simple* graph² of the given degree sequence has equal probability of being generated. An exact condition is known in terms the given degree sequence for the model to possess a giant component [56], the expected size of that component is known [57], and the average size of non-giant components both above and below the transition is known [63]. Another model is Chung-Lu’s model [21] that creates an edge between a pair of nodes with probability proportional to the product of their given degrees. The *expected* degrees of nodes in the generated graph equal the given degrees. Similarly to the work on the configuration model, many basic structures of random graphs generated by this model have been studied [20, 21]. A disadvantage of Chung-Lu’s model is its constraint on the maximum given degrees, since in order to make the values of probabilities meaningful so that they lie in the interval $[0, 1]$, the square of the maximum given degree should be less than or equal the number of edges. Park and Newman [64] deal with this issue and study a generalize Chung-Lu’s model.

2.3 Dynamics and Algorithms on Social Networks

In 1960s, Stanley Milgram did his famous “small-world” experiments [55, 69]. The following description of the experiments can be found in [60]. The experiments probed the distribution of path lengths in an acquaintance network by asking participants to pass a letter to one of their first-name acquaintances in an attempt to get it to an assigned target individual. Most of the letters in the experiment were lost, but about a quarter reached the target and passed on average through the hands of only about six people in doing so. This experiment was the origin of the popular concept of “six degrees of separation,”

²A graph is simple if there is at most one edge between each pair of vertices and there is no self loops.

although that phrase did not appear in Milgram's writing, being coined some decades later by Guare [43]. A brief but useful early review of Milgram's work and work stemming from it was given by Garfield [40].

To be a “small-world”, a network should also have a high clustering coefficient (also called transitivity), see Definitions 1.1.9 and 1.1.10. Usually the two parameters, (short) lengths of shortest paths and (high) values of clustering coefficients are competing with each other in sparse graphs, e.g., social networks. The influential work of Watts and Strogatz [72, 73, 74] reconciled these by positing a network built on a low-dimensional regular lattice and then adding or moving edges to create a low density of “shortcuts” that join remote parts of the lattice to one another. The rewiring process allows the small-world model to interpolate between a regular lattice and something which is similar, though not identical, to a random graph. As Watts and Strogatz showed by numerical simulation, there exists a sizable region in between these two extremes for which the model has both low path lengths and high clustering coefficients.

As described in [60], Kleinberg [48, 49] pointed out that the results of Milgram's famous small-world experiment not only showed that there exist short paths through social networks between apparently distant individuals in the social network, but they also demonstrate that ordinary people are good at finding them. The latter conclusion was apparently not noticed by Milgram. This is perhaps an even more surprising result than the existence of the paths in the first place. The participants in Milgram's study had no special knowledge of the network connecting them to the target person. Most people know only who their friends are and perhaps a few of their friends' friends. Nonetheless it proved possible to get a message to a distant target in only a small number of steps. This indicates that there is something quite special about the structure of the network. On a

random graph for instance, as Kleinberg pointed out, short paths between vertices exist but no one would be able to find them given only the kind of information that people have in realistic situations.

Kleinberg’s contribution [48, 49] is perhaps the pioneering work of applying rigorous algorithm design and analysis to social networks studies. There are also some other problems in social and complex networks that have been modeled as algorithm and graph theory problems. For instance, the network resilience problem studies how resilient a network is to random or targeted deletion of their vertices [4, 17, 18, 22]. Bollobás and Riordan [16] analyzed this problem in a mathematically rigorous way. Usage of social networks as a medium for the spread of information, ideas, and influence among its members has long been exploited in practice. In recent work, motivated by applications to marketing, Domingos and Richardson posed a fundamental algorithmic problem for such systems [25, 26]. Kempe, Kleinberg and Tardos [47] studied the issue of choosing influential sets of individuals as a problem in discrete optimization, and obtain the first provable approximation guarantees for efficient algorithms in a number of general cases.

Our work follows the same line of applying algorithms design and analysis to the modeling of social networks. In particular, we design combinatorial formulations for modeling the problems of controlling epidemics, and develop efficient approximation algorithms to tackle the problems of stopping or slowing down the spread of disease.

Chapter 3

TRANSIMS and EpiSims

3.1 TRANSIMS

In order to understand the spread of contagious diseases, we need a realistic representation of a social contact network. The TRansportation ANalysis and SIMulation System (TRANSIMS) [10] developed at Los Alamos provides a way to generate synthetic realistic social contact networks in a large urban region. It is based on the assumption that the transportation infrastructure constrains people’s choices about what activities to perform and where to perform them. TRANSIMS produces positions of all travelers on a second-by-second basis in a large metropolitan area and it has effectively been used to construct detailed mobility patterns for the city of Portland. We refer the reader to [10] and the web-site <http://transims.tsasa.lanl.gov> for more extensive descriptions of this tool. TRANSIMS conceptually decomposes the transportation planning task using three different time scales, as follows.

(1) *Creating a population and activities.* Data about land use and demographic information, combined with survey data from thousands of households is employed to create a synthetic population, where each person has a specific home address. A sequence of daily activities, and the locations where these activities are to be done, is determined for each person, based on the activity surveys, travel time and land use data.

(2) *Assigning Routes and trip-chains.* Second, an intermediate time-scale consists of assigning routes and trip-chains to satisfy the activity requests. To do this, the estimated

locations are input to a routing algorithm to find minimum cost paths through the transportation infrastructure consistent with constraints on mode choice [9, 11]. An example of constraints might be: “walk to a transit stop, take transit to work using no more than two transfers and no more than one bus.” This step is coupled with the simulation of the actual movement of people on their chosen routes, and is repeated till some sort of near-equilibrium is attained.

(3) Detailed simulation. Finally, the movement of people along their chosen routes is simulated. This simulation is extremely detailed: it resolves distances down to 7.5 meters and times down to one second. It provides an updated estimate of time-dependent travel times for each edge in the network, including the effects of congestion. This estimate is input back to the Routing (step (2)) and location estimation (step (1)) algorithms, which produce new plans. This feedback process continues iteratively until it converges to a “quasi-steady state” in which no one can find a better path in the context of everyone else’s decisions.

A substantial effort has been spent on calibration and validation of the output produced by TRANSIMS, and it has been deployed in the city of Portland (see [10] for details). Various microscopic and macroscopic quantities produced by TRANSIMS have been verified in the city of Portland at a statistical level; these include (i) traffic invariants such as flow density patterns, jam wave propagation, (ii) macroscopic quantities, such as activities and population densities in the entire city, the number of people occupying various locations in a time varying fashion, time varying traffic density split by trip purpose and various modal choices over highways and other major roads, turn counts, number of trips going between zones in a city, etc. TRANSIMS produces a comprehensive representation of people and their activities over the course of a day. Figure 3.1 shows an example of

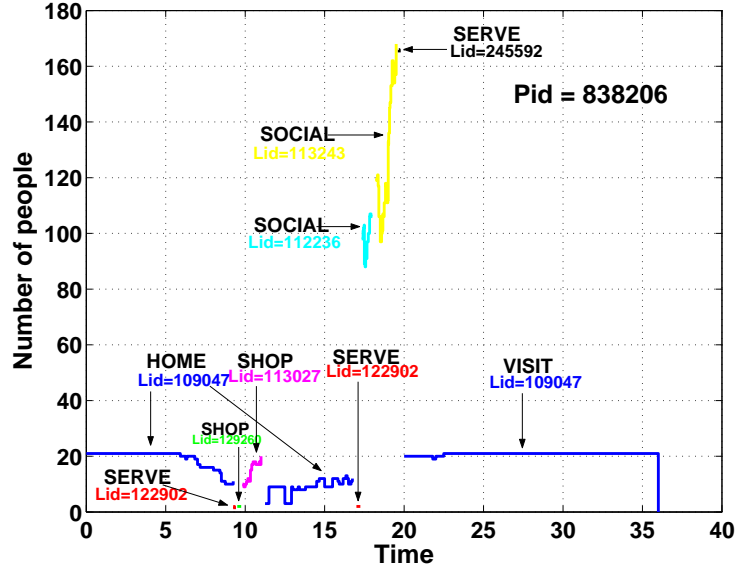


Figure 3.1: Figure showing, for a randomly chosen synthetic individual (whose person-id is $\text{pid}=838206$ as shown in the figure) in Portland, the activities, their locations (marked with lid's as their location-id's) and their time durations (in hour) over the course of 1.5 days. The x -axis indicates the time when an activity happens, the y -axis indicates the number of people in the same location where the activity is taken by the chosen individual. In Portland data there are nine activity types, namely **home**, **work**, **shop**, **visit**, **social**, **other**, **serve**, **school**, **college**. Each edge in the people-location bipartite graph is labeled by one of these activities.

this for a randomly chosen synthetic individual. It is important to note that simulations such as TRANSIMS appears to be the *only way* to obtain such detailed information about some of the measures discussed here.

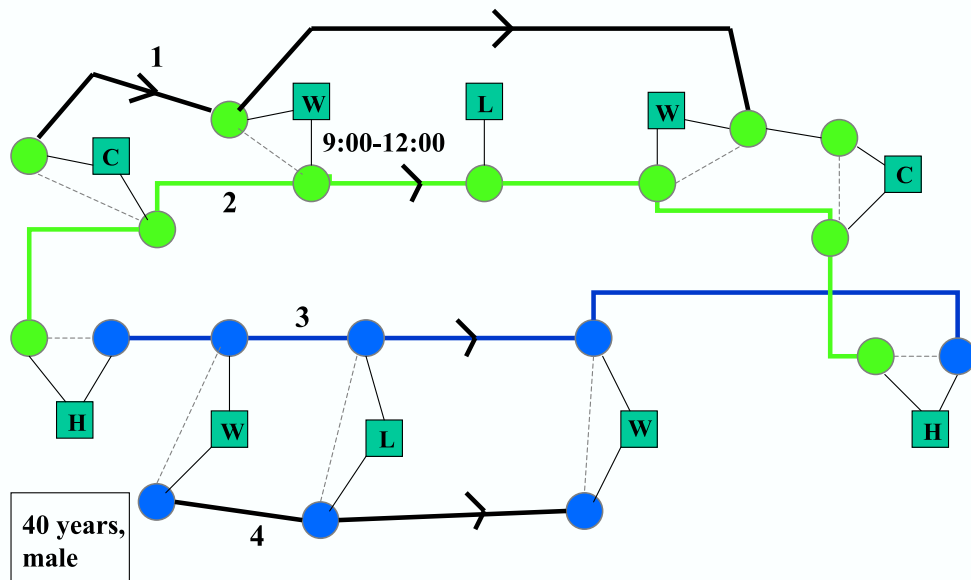


Figure 3.2: An illustration of various steps in TRANSIMS, and the networks that are constructed. The squares denote locations, with the letters specifying the type of locations (H - home, W - work, C - carpool, L - place for lunch). The circles denote people, moving from one location to another. The thin solid lines between people and locations show the edges between people and locations, i.e., the edges of the graph G_{PL} – these edges are labeled with the time duration when the person is present at the location. The thick solid lines (with the arrows) show the “trajectory” of a person (four persons in this figure), as they move from one location to another; this also illustrates the temporal aspects of the network – for instance, person-1 moves from home to a carpool (viewed as a location), to a work location, and so on. At a location, all people who are at that location at the same time are connected by dashed edges - these are the edges of the people-people contact graph. The people also have labels – one such label (40 years old, male) is shown in the figure.

In summary, TRANSIMS generates the following information for a city – demographic information for each person and location, and a minute-by-minute schedule of each person’s activities and the locations where these activities take place. This information can be abstractly represented by a (vertex and edge) labeled bipartite graph G_{PL} , where P is the set of people and L is the set of locations. If a person $p \in P$ visits a location $\ell \in L$, there is an edge $(p, \ell, label) \in E(G_{PL})$ between them, where *label* is a record of the type of activity of the visit and its start and end time. Each vertex (person and location) can also have labels. The person labels correspond to his/her demographic attributes such as age, income, etc. The labels attached to locations specify the location’s attributes such as its x and y coordinates, the type of activity performed, maximum capacity, etc. Note that, there can be multiple edges between a person and a location recording different visits. When studying the topological structure of contact networks, we will sometimes consider and sometimes ignore the labels and the multiplicity of the edges; see [29] for a discussion of why the time-labels can be ignored for various diseases. We use the term *people-location* graph to refer to the above bipartite graph, wherein multiple edges are discarded and time labels are omitted.

We also consider another graph G_P induced on the set of people: $(p_1, p_2) \in E(G_P)$ if there is a location $\ell \in L$ such that $(p_1, \ell), (p_2, \ell) \in E(G_{PL})$, and the time intervals during which p_1 and p_2 are present at ℓ overlap, i.e., there is a common location at which the two people p_1, p_2 are present at the same time. This graph will be referred to as the *people-people* graph. In this thesis, all the data is for the city of Portland, Oregon, USA; our ongoing research is studying the (broadly similar) structures of other urban areas such as Chicago.

3.2 EpiSims

EpiSims is a tool for simulating the spread of disease on a social contact network. We now provide a brief overview of this tool; further details can be found in [32, 33]. Using the information generated by TRANSIMS, the simulation maintains a parameterized model for the state of the health of each person, and updates this continuously based on interaction with other people, and transmission of the disease through these contacts. This enables us to get estimates of both the geographic and demographic distribution of disease as a function of time; it also allows us to evaluate the impact of different vaccination/quarantining policies. Different aspects of this tool are discussed below.

(1) *Disease Model within each Host:* EpiSims uses a single parameter, the *disease load*, to represent the effect of a disease upon a host. The load in EpiSims is intended to be analogous to viral titre in a throat swab, number of spores or bacteria present, concentration of toxin, etc. However, it need not reproduce such clinical aspects of these loads as distribution throughout the body. It is merely a parameter that is used to determine whether a person is infected, symptomatic, too sick for normal activities, infectious, or dead - the higher the disease load, the sicker the person. Each individual in the simulation who is exposed (either through exposure to an initial release or through contact with an infected and infectious person) will progress through a series of disease stages. An exposed individual will either become infected or not with a probability based upon the disease model and the individual's demographics. Individuals who become infected either develop a clinical case of the disease or not. For instance, some fraction of those infected with smallpox never develop a fever or symptoms of the disease. As above, the probability of developing clinical symptoms depends upon individual characteristics. An isolated con-

taminated person's or location's load grows or shrinks at predetermined rates, depending on the characteristics of the person and location. All locations share a single common exponential growth or decay rate, depending on the disease. For example, the amount of virus present in the environment would decay exponentially if there were no sources (infected people); the amount of bacteria might grow exponentially; while the number of spores would remain fixed. The disease model also specifies a set of threshold values for determining the effect of the load on an individual; these thresholds determine whether an individual is infected, symptomatic, infectious, dead, etc.

(2) *Disease transmission and progression:* In EpiSims, an infectious person contaminates his or her environment, in a process analogous to sneezing or coughing. The contamination may be restricted to a small region near the infected person, and/or it may spread to an entire location, which is roughly the size of an apartment building, office building, or shopping mall. Transmission occurs as uninfected people absorb virus (or bacteria, spores, etc.) from a contaminated location.

Geographical locations, as well as people, have a disease load associated with them, representing the level of contamination of the location. Disease load in a location has an associated exponential growth rate, which may be positive, negative, or zero. This allows EpiSims to model non-infectious diseases, transmission of disease between people who are never in direct contact, or diseases with non-human vectors. The simulation can be initialized by contaminating a specific location at a specific time and/or by assigning a non-zero load to one or more people.

There are two corresponding parameters controlling the interaction of each person with his or her local environment: the *shedding rate*, the fraction of the individual's load that is shed to the environment per hour, and the *absorption rate*, the fraction of the

environment's load that is absorbed by an individual per hour. These parameters are specific to the individual, and can be set from an estimate of how long a person must be in close contact with an infectious person before becoming infected.

Chapter 4

Structures of Portland Graph and Strategies to Minimize Disease

Spreading

In this chapter we study basic structural properties of the social contact network for Portland. By cultural properties, we primarily mean demographic analysis of locations and people, especially as it is constrained by their interaction. In addition to standard measures, such as degree and clustering coefficient distributions, we identify new measures which are more relevant to disease dynamics: overlap ratios, expansion, shattering, temporal degrees and demographic mixing rates. As we argue later, the overlap ratio is a more useful measure than the clustering coefficient. We also explain the significance of expansion and shattering for network epidemiology. The temporal degrees and demographic mixing rates have not been studied previously at this level of detail (due to the lack of sufficient data), and can be used for developing new vaccination policies.

Based on these structural measures, we show via simulating vaccination and quarantine by shattering the network, that vaccination and quarantine alone are not a practical way to stop the disease spreading, due to the expansion properties of the social network. At the end, we exploit a favorable structure of the Portland data, overlap ratios, to show that early detection of disease is feasible in this social network, and if combined with vaccination and quarantine, could result in efficient strategies against disease spreading.

4.1 Basic Structure

In the bipartite graph G_{PL} (see Section 3.1 for definitions) for Portland, we have 1615860 ($\tilde{1.6}$ million) people, 181230 ($\tilde{181K}$) locations, and 6060679 ($\tilde{6.1}$ million) edges. Figure 4.1 shows the degree distributions of the locations and people in the bipartite graph G_{PL} for the Portland data. A large part of the degree sequence of locations follows a Power-law distribution, i.e., $n_k \propto k^{-\beta}$, where n_k denotes the number of locations of degree k ; for the Portland data, $\beta \approx 2.8$. The degree sequence of people is roughly Poisson distributed with maximum degree 15.

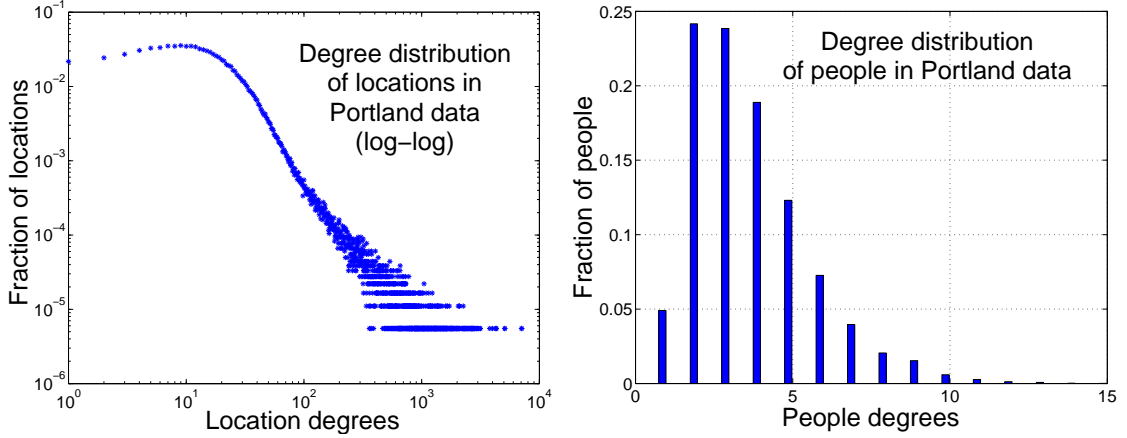


Figure 4.1: Degree distributions of locations and people in the bipartite graph G_{PL} for Portland data. The location degrees range from 1 to 7091, people degrees range from 1 to 15.

The degree sequence of people in G_P (see Section 3.1) is shown in Figure 4.2, and looks quite different than the degree sequence of G_{PL} . The graph G_P for Portland is not fully connected, but has a giant component with 1615813 people (almost all the people). We also determine the numbers of length-2 paths and triangles: a length-2 path is an unordered pair of edges $((a, b), (b, c))$ that share one vertex; a triangle is an unordered triple of vertices (a, b, c) such that each of the three pairs is an edge. Letting

Δ denote the number of triangles and Γ_2 denote the number of length-2 paths, the global clustering coefficient is defined as $3\Delta/\Gamma_2$ (see Definition 1.1.10). For G_P of Portland data, $\Delta \simeq 6.3117 \times 10^{11}$ and $\Gamma_2 \simeq 3.349 \times 10^{12}$; the global clustering coefficient for the Portland data is about 0.565268. Recent literature on social networks (see e.g. [8, 59]) has given a lot of importance to this measure, suggesting that a large clustering coefficient implies a more “tight-knit” interaction; however, we argue later that the overlap ratios and expansions are much better measures in this regard. It is worth noting that although we haven’t found a general algorithm to count the number of triangles in a graph in linear time, it is quite easy to count the number of length-2 paths in linear time. Lemma 4.1.1 shows that the number of length-2 paths can be computed from the degree sequence.

Lemma 4.1.1 *Given the degree sequence $\{(d_i, n_i)\}$ where d_i ’s are different degrees and n_i is the number of vertices having degree d_i , the number of length-2 paths is*

$$\Gamma_2 = \sum_{d_i \geq 2} \frac{d_i(d_i - 1)}{2} \cdot n_i. \quad (4.1)$$

Proof If a vertex’s degree is less than 2, it cannot be the center of any length-2 path.

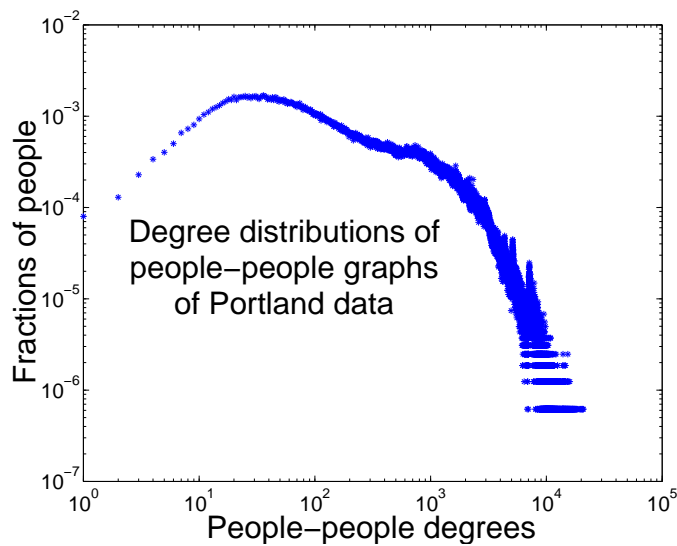


Figure 4.2: Degree distribution in the people-people graph for the Portland data.

For a vertex v of degree $d_i \geq 2$, the number of length-2 paths whose center point is v is $\binom{d_i}{2} = \frac{d_i(d_i-1)}{2}$. Since their center points form a one-to-one mapping to the length-2 paths, Formula 4.1 computes the total number of length-2 paths in a graph. \square

Another commonly used definition of clustering coefficients is shown in Definition 1.1.9, where the neighborhood of each vertex has a clustering coefficient and their average value is the average clustering coefficient for the graph. It is worth noticing that although compute the clustering coefficients for all vertices in a graph is doable, for a graph as large as the Portland data, it is not practical to compute all of them just in order to obtain the average clustering coefficient of the graph. In fact, we can uniformly sample $\Theta(\log |P|)$ of the people and compute their average clustering coefficient as an estimate to the one of the whole graph G_P . By using the Chernoff bound (Lemma 1.1.7) we can easily prove that, with probability $1 - O(\frac{1}{|P|})$, the sampled average clustering coefficient is within $1 \pm \epsilon$ of the actually average clustering coefficient, where $0 < \epsilon < 1$ is a small positive constant. We sampled about 70 vertices in G_P and computed their average clustering coefficient which is 0.6376. Although it is not equal to the global clustering coefficient 0.565268 shown above, we note that these two clustering coefficients are not exactly the same by definition, but they both capture the nature of the local density of a social network. In the rest of this thesis, we only consider the global clustering coefficient.

Length of shortest paths is also an important measure of social networks. Along with high clustering coefficients, short average shortest paths illustrates that the underlying graph is a small-world graph. To compute the lengths for all pairs of vertices in the graph is not practical for a social network as large as Portland data. We again use sampling methods to estimate not only the average length of all shortest paths, but also the distribution of lengths of them. There are two kinds of sampling methods. One is to

	len=1	len=2	len=3	len=4	len=5	len=6
mean	8.0806×10^{-4}	0.2666	0.7126	0.0200	4.8687×10^{-5}	5.6341×10^{-9}
std.	9.1226×10^{-5}	0.0149	0.0138	0.0049	8.8549×10^{-5}	9.8875×10^{-9}

Table 4.1: Means and standard deviations of fractions of shortest paths of different lengths in the shortest-path spanning trees sampled from the giant component of Portland’s G_P graph.

independently sample some *pairs* of vertices and compute their shortest path. Another one is to independently sample some vertices and for each one of them, compute the *shortest path spanning tree* rooted at it. The path from the root to each vertex in the tree is a shortest path between them. Although the sampled paths are not independent of each other, as long as we only consider the paths of lengths not exceeding a certain value, the tail bound from [66] ensures that only a small number of trees are needed to estimate the shortest path distribution. In realistic social networks and in Portland data, almost all shortest paths are within a small range that permits a small number of sampling. The running time of computing a shortest path between a pair of randomly chosen vertices is roughly the same as that of computing the whole shortest path spanning tree rooted at a randomly chosen vertex. On the other hand a shortest path spanning tree simultaneously gives the $|P| - 1$ shortest paths between the root and all other $|P| - 1$ vertices, using almost the same running time as computing *one* shortest paths between a randomly chosen pair. Therefore the second method gives us much more samples within roughly the same time than the first method does. We sampled about seven hundred vertices in Portland’s people-people contact graph G_P and computed the shortest paths spanning trees rooted at them. The statistics of the distribution of shortest paths based on these samples is shown in Table 4.1.

4.2 Graph Expansion and Shattering

We consider the two standard notions of expansion in the graph G_P . The edge expansion of a subset $S \subseteq P$ is defined as the ratio

$$\frac{|\{e = (u, v) : (u, v) \text{ is an edge and } u \in S, v \notin S\}|}{|S|}.$$

The vertex expansion of a subset $S \subseteq P$ is defined as the ratio $|\{u \notin S : (u, v) \text{ is an edge and } v \in S\}|/|S|$. The edge (respectively vertex) expansion of G_P is the minimum, taken over all $S \subset P$ such that $|S| \leq |P|/2$, of the edge (respectively vertex) expansion of S . The vertex and edge expansions are important graph-theoretic properties that capture fault-tolerance, speed of data dissemination in the network, etc. Roughly, the higher the expansion, the quicker the spread of any phenomenon (disease, gossip, data etc.) along the links of the network. Computing the expansion exactly is NP-hard, but can be approximated within a polylogarithmic factor using the results of Leighton and Rao [52]. However, the algorithm of [52] is currently unsuitable due to its high computational cost for analyzing large graphs such as the ones studied here; hence we use random sampling to estimate the vertex and edge expansions.¹ We collected approximately 500,000 random samples of subsets of different sizes and calculated the smallest vertex and edge expansion, among all samples. Figure 4.3 summarizes the results. The Y-axis plots the smallest expansion value found among the 500,000 independent samples; the X-axis plots the set size S as a percentage of the total number of vertices in the graph (the sampling probability). The plots labeled “Vertex expansion-2” and “Edge expansion-2” in Figure 4.3 show the expansion in the graph G_P , while the plots marked “Vertex expansion-1” and “Edge expansion-1” show the same quantity on a sparser people-people graph – the graph is made sparser by only

¹Random walk based methods have been used in property-testing type of algorithms for determining expansion (see e.g. [41]).

retaining edges between individuals who came in contact for at least one hour. It is evident that the expansion rate does go down in the sparser graph; nevertheless, both the plots show a very high expansion rate.

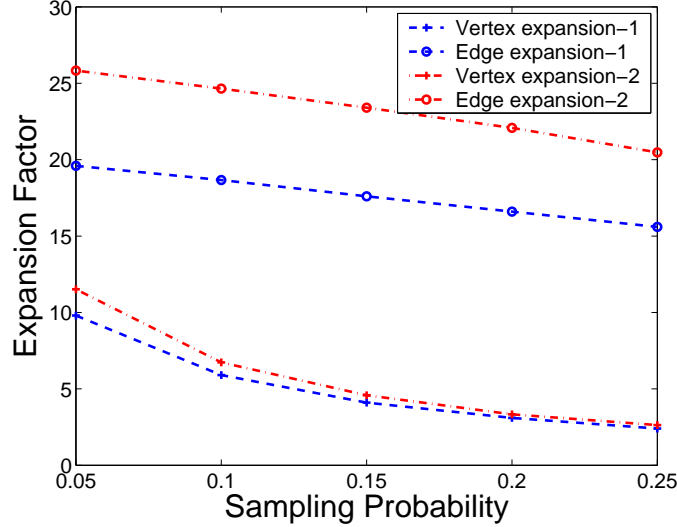


Figure 4.3: Expansion of the people-people graph: the plots marked “Vertex expansion-2” and “Edge expansion-2” show the vertex and edge expansion for the graph G_P , while “Vertex expansion-1” and “Edge expansion-1” show the corresponding quantities in the graph obtained by retaining only those edges that involve an interaction of at least 1 hour. This leads to a much sparser graph and correspondingly lower values of vertex and edge expansions.

The high expansion implies that contagious diseases would spread very fast, and makes early detection imperative, in order to control the disease. We discuss detection strategies later in Section 4.3. Recent papers, such as [3, 17, 18, 61], have proposed strategies such as vaccination of high-degree people. In the light of the high expansion, such strategies are unlikely to be very effective. To quantify the effectiveness of such a vaccination policy, we study a natural measure called *shattering* – given a parameter $\alpha > 0$, this corresponds to the minimum size of the largest connected component, over all possible ways of removing up to αn nodes. Vaccinating an individual can be viewed as removing all incident edges on this node (since the individual will no longer contract the disease

and further transmit it). A vaccination scheme (corresponding to node deletions) that leads to small connected components implies that the disease would not spread beyond any component. Therefore, the size of the largest connected component in the graph resulting from the deletion of all “vaccinated” nodes is a measure of the effectiveness of the vaccination scheme. In other words, it is desired that vaccination should *shatter the graph* into small components. Figure 4.4 shows the sizes of the largest components after repeatedly removing nodes from the largest degree to the smallest degree. From the figure one can see that the largest component remains very stable, and continues to remain unique until all nodes of degree 11 are deleted, which requires deleting a large fraction (0.698) of the nodes. This suggests that for contagious diseases, the “high degree node” heuristic is in practice no different than mass vaccination.

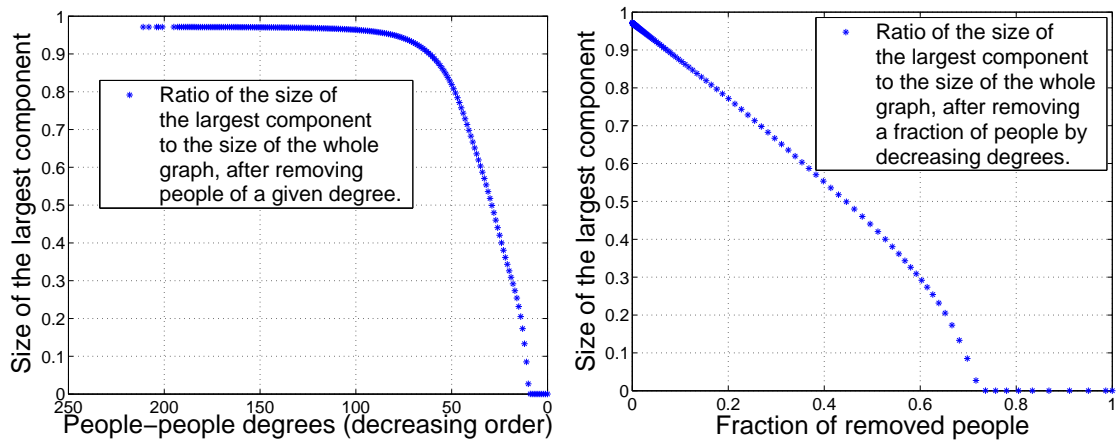


Figure 4.4: Size of the giant connected component after each iteration.

4.3 Overlap Ratios and the Sensor Placement Problem

As discussed earlier in Section 4.2, the expansion of the social network is very high, leading to high rates of spread of diseases, and making simple targeted vaccination schemes infeasible. This makes the problem of early detection even more important. One way of

detecting certain kinds of diseases (such as small pox and anthrax) is to place sensors in some public places. This leads to the following problem: choose a subset $L' \subseteq L$ of locations to place the sensors, so that all or most people visit at least one of these locations; the goal is to choose a set L' of the smallest size.

We note that the sensor placement problem has been discussed elsewhere for different problems; for example, in studying municipal water networks [14]. In our situation, the sensor problem reduces to the classical dominating set problem (see Problem 1.1.3). It is NP-hard, and a greedy algorithm gives an $O(\log |L|)$ approximation (see Algorithm 1.1.5). Furthermore the running time of this algorithm is not efficient for large data-sets. For the G_{PL} graph of Portland data, a particular structure, the overlap ratio, hints that there may exist more efficient algorithms.

For a set $L' \subseteq L$ of locations, let $N(L')$ be the set of individuals visiting at least one of the locations in L' , i.e., $N(L') = \{p \in P : (p, l) \in E(G_{PL}) \text{ and } l \in L'\}$. Then the overlap ratio of L' is defined by $\frac{|N(L')|}{\sum_{l \in L'} d(l)}$. For a given integer d , the Point-Overlap-Ratio(d) is the overlap ratio defined for the set of locations of degree d , and the Cumulative-Overlap-Ratio(d) is the overlap ratio of the set of all locations having degree at least d . The first two plots in Figure 4.5 show the overlap ratios of Portland data for the sets of locations considered in decreasing order of degrees. The plots show that high-degree locations are visited by *almost-disjoint* sets of individuals. In other words, most pairs of distinct high-degree locations are visited by (almost) disjoint sets of individuals during a day. This is an important structural property of the social network under investigation and is likely to be true for many other such social networks. This has important implications from the standpoint of designing effective strategies for monitoring the onset of an epidemic.

Based on the above observation, we designed a much simpler algorithm FAST-

GREEDY [30] running in linear time. Instead of finding the highest degree location after each iteration, the FASTGREEDY algorithm sorts the locations by their degrees in advance, and chooses locations one-by-one from the highest degree down to the lowest degree until the chosen locations dominate a required fraction of people. The formal description of FASTGREEDY is in Algorithm 4.3.1.

Algorithm 4.3.1 (FASTGREEDY)

1. *Sort the locations in L in non-increasing order of their degrees, i.e., $d(\ell_1) \geq d(\ell_2) \geq \dots \geq d(\ell_{|L|})$.*
2. *Select the smallest i^* such that $|\bigcup_{j \leq i^*} N(\ell_j)| \geq (1 - \epsilon)|P|$, as follow: repeatedly choose locations from the highest degree to the lowest degree until $(1 - \epsilon)|P|$ people have been dominated*
3. *Take the subset $\{\ell_1, \dots, \ell_{i^*}\}$ as the dominating set (dominating $(1 - \epsilon)$ -fraction of the people).*

By a closer examination of Algorithm 4.3.1 we can see that some locations whose people have already been dominated by previously chosen locations don't need to be in the dominating set. Therefore a frugal version the FASTGREEDY is presented below. Note that even though the performance is improved, the running time of the frugal FASTGREEDY algorithm is still almost the same as the FASTGREEDY algorithm.

Algorithm 4.3.2 (Frugal-FASTGREEDY)

1. *Sort the locations in L in non-increasing order of their degrees, i.e., $d(\ell_1) \geq d(\ell_2) \geq \dots \geq d(\ell_{|L|})$.*

2. Let $\mathcal{D} \leftarrow \emptyset$. Repeat the following process until $(1 - \epsilon)|P|$ people have been dominated:
Extract the first location ℓ in the ordering; if any people dominated by ℓ have not been dominated by \mathcal{D} , let $\mathcal{D} \leftarrow \mathcal{D} \cup \{\ell\}$.
3. Take \mathcal{D} as the dominating set (dominating $(1 - \epsilon)$ -fraction of the people).

Figure 4.5 shows that the FASTGREEDY heuristic works very well in practice. It performs almost as well as the Frugal-FASTGREEDY and the classical fast greedy algorithm (see Algorithm 1.1.5), to dominate up to 95% of people. It is worth noting that Algorithm 1.1.5 uses 41.23% of locations to dominate all people and the Frugal-FASTGREEDY uses 47.35% of locations to dominate all people. The running time of Algorithm 1.1.5 is, however, four hours, while the running time of the Frugal-FASTGREEDY is less than 15 seconds for Portland data. The effectiveness of FASTGREEDY is, intuitively, due to the high overlap ratios. We will rigorously prove in Chapter 5 that for Chung-Lu’s model and the configuration model, FASTGREEDY is a $(1 + \epsilon)$ -approximation algorithm to the $(1 - \epsilon')$ -dominating set problem, with high probability.

4.4 Other Structural Properties

Network analysis has typically only dealt with static graphs, but real graphs are dynamic: even simple measures such as degree distributions are temporal functions. Figure 4.6 shows the temporal variation in degrees for six different location types in Portland; for each location type, the distribution of four randomly chosen locations is shown. The degree distributions reflect the basic trend that one expects; for example, the number of individuals at home is high in the early morning hours, decreases during the day and then shows an increase during evening hours. Work locations, in contrast, show a complementary behavior. Instead of “high degree” type of vaccination strategies, one can propose

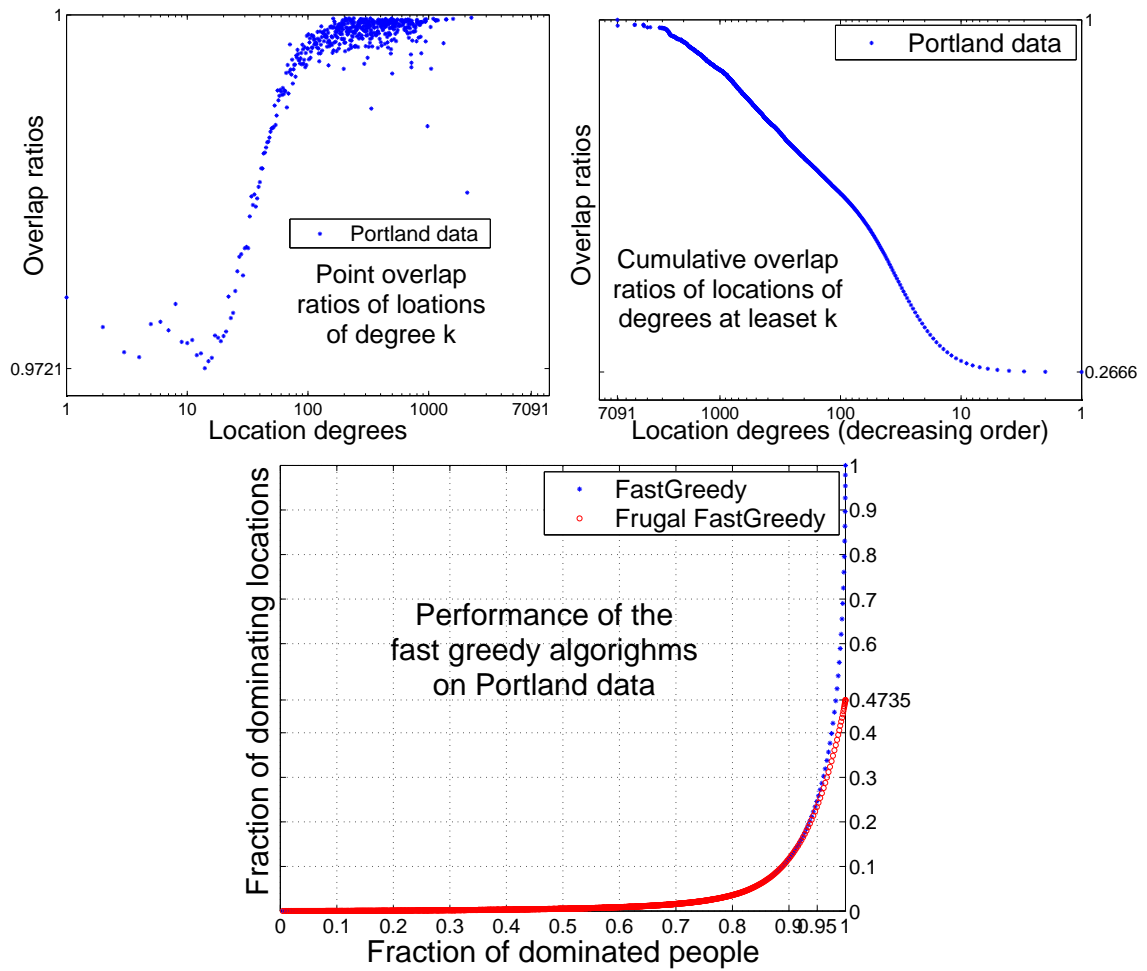


Figure 4.5: Overlap ratios and performance of the fast greedy algorithms for the dominating set problem (Portland data).

much more refined schemes by using the temporal degrees.

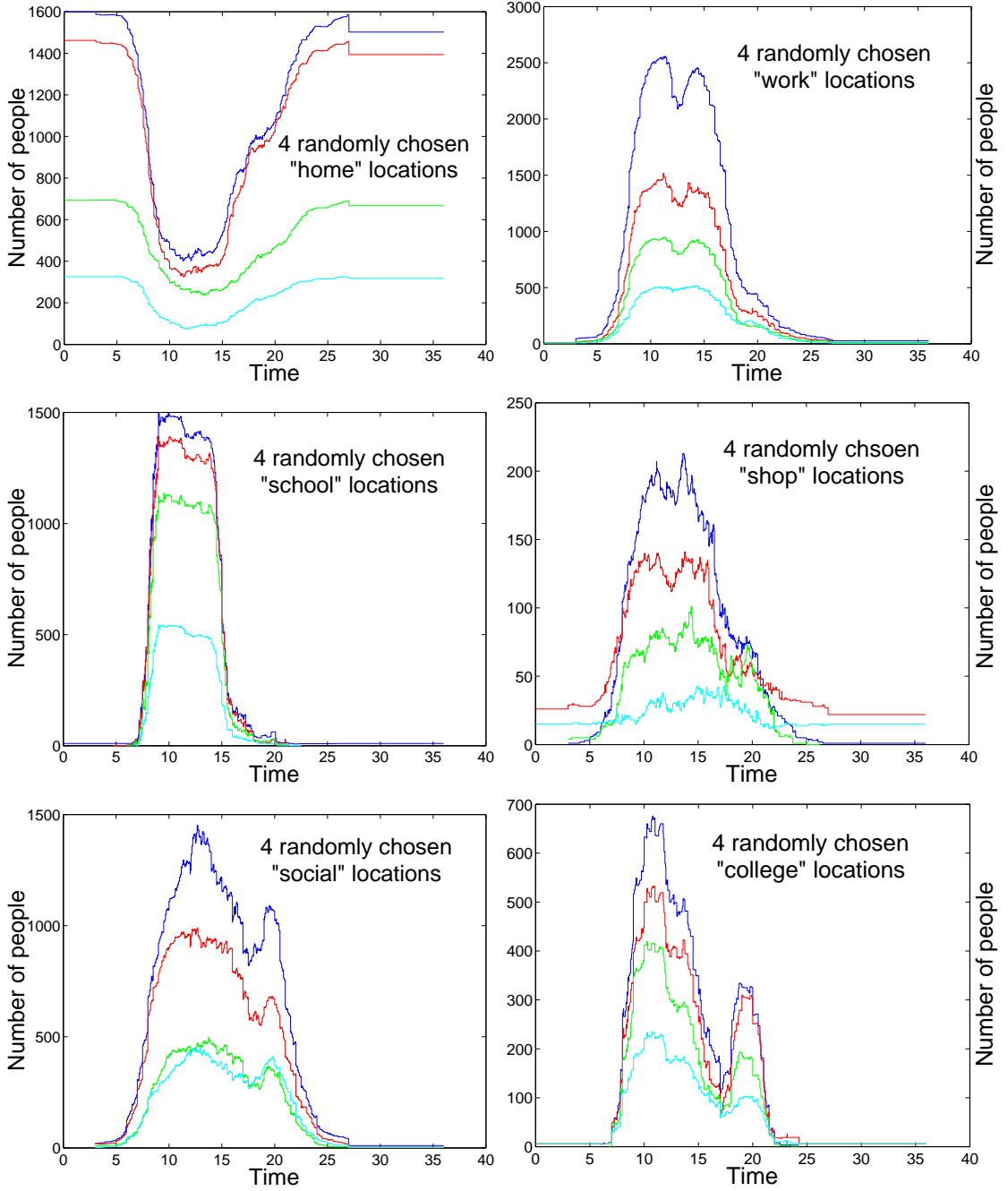


Figure 4.6: Temporal degree distributions for different types of activity locations. The types of activity locations are: `home`, `work`, `school`, `shop`, `social`, `college`. Each plot contains the temporal degree of four randomly chosen activity locations of a given type, where the x -axis shows the time in hours, and the y -axis shows the number of people at that time (x) and that location (doing the specific activity). For example, the top left plot shows the temporal degree distribution for four randomly chosen home locations. The home location is really a block of homes, and that accounts for the large sizes.

We also study the activity statistics of Portland data. The activity statistics are shown in Figure 4.7.

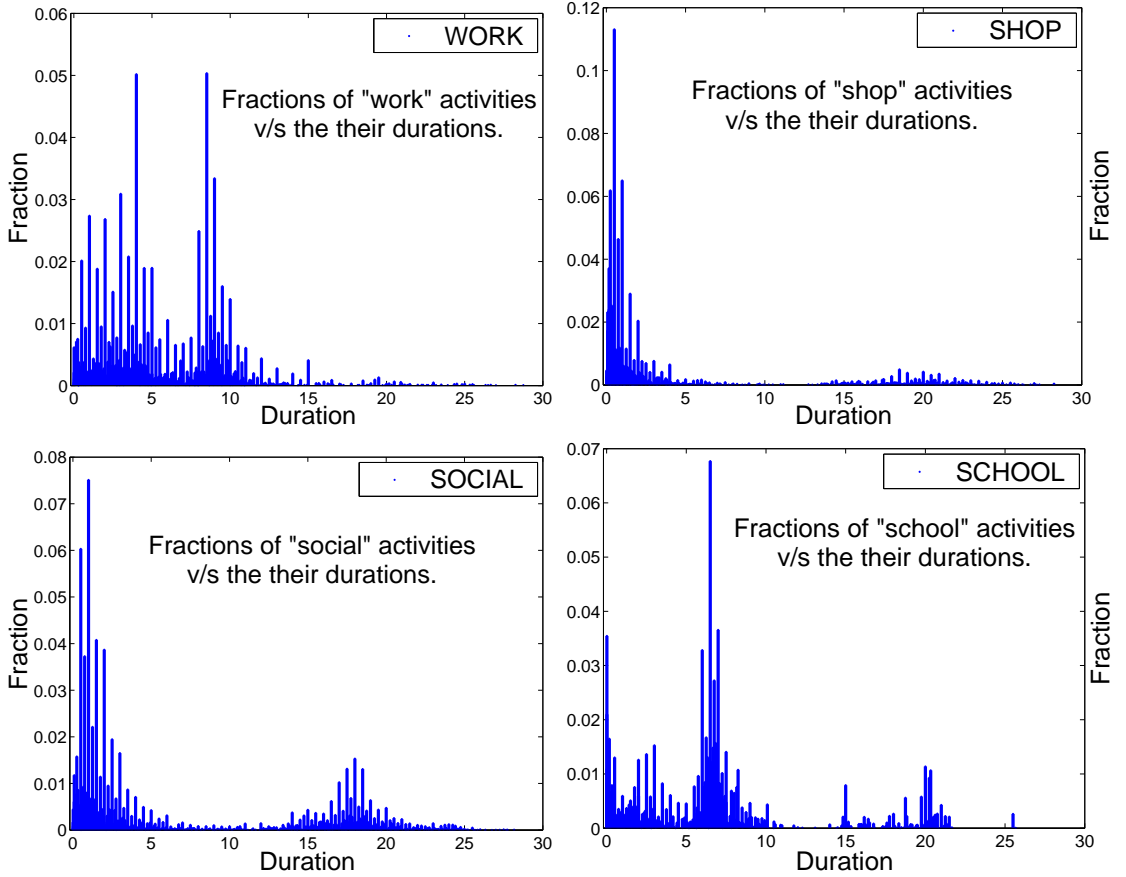


Figure 4.7: Distribution of activity lengths for *work*, *shop*, *social*, *school* activities. For each activity type, and for each possible duration of times d , the plot shows the fraction of this activity lasted for d time units.

As described earlier in Section 3.1, the data from TRANSIMS also contain information related to the type of activity a person does at a location; the types of activities range from *home* and *work* to *school* – this describes the kind of activity done at the location. Figure 4.7 shows the distributions of lengths of some activities, obtained from the temporal G_{PL} graph, and shows some interesting features. For instance, consider the leftmost panel in Figure 4.7, which corresponds to the activity type *work*. There are two peaks in the distribution: the first peak is around 4 hours and the second is around 8 hours;

these peaks correspond well with our intuition of half and full work days. Activity length information could also be useful for decision making: for instance, in contact tracing, it might suffice to ignore activities that had much smaller duration.

Unlike in uniform mixing models, contacts between people are very non-uniform, and depend on a host of demographic factors. For instance, teens have more contacts with other teens and with some 30-40 years old (probably parents, teachers) than with people of other age groups, as the first plot in Figure 4.8 shows. To capture this heterogeneity in the contacts, we consider people of different age-groups, and determine the average number of contacts with people of other ages; Figure 4.8 shows the distributions for three specific age-groups of 16, 30 and 60 (see [29] for more details). Such demographic mixing information can be used in determining refined vaccination policies.

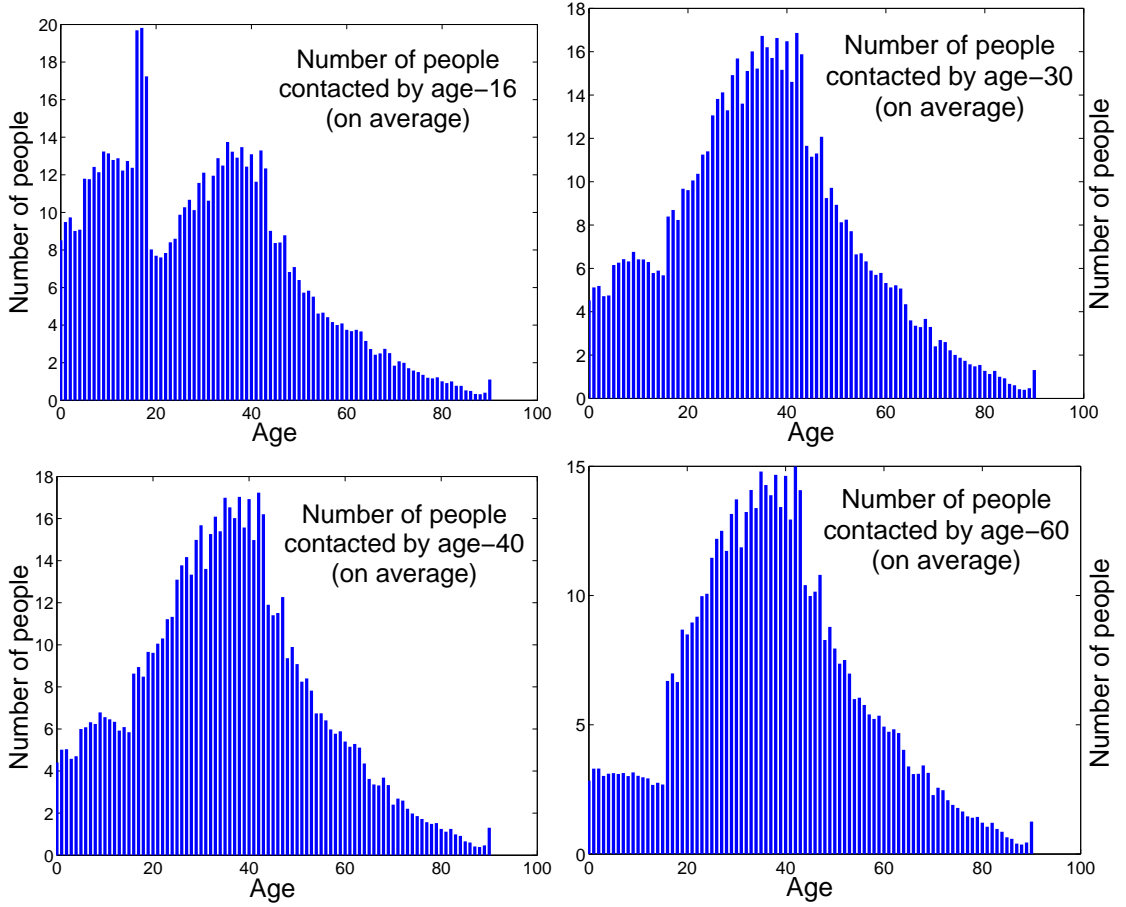


Figure 4.8: Contacts with each age-group, for people of ages 16, 30, 40, and 60. For each of these age groups (say A), and for each possible age-group (say B) on the X -axis, the plot shows the average number of contacts that group A makes with group B in the Y -axis. The average for a given age is computed by computing a distribution for each person p of that age and then summing up these distributions and dividing the resulting values by the total number of individuals of that age.

Chapter 5

Random Graphs Models For Social Networks

In Chapter 4 we have shown empirical results of the sensor placement problem (i.e., the dominating set problem see Problem 1.1.3). We showed that FASTGREEDY (see Algorithm 1.1.5) performs very well on Portland data. The dominating set problem is, however, NP-hard, and we don't know what the optimum solution is on Portland data. In order to show that FASTGREEDY indeed returns solutions close enough to the optimum one, we use random graph approaches (see, e.g., [5]) to prove that in Chung-Lu's and the configuration models, with high probability, FASTGREEDY is a $(1 + \epsilon)$ -approximation algorithm to the dominating set problem for networks resembling Portland data.

Since our proof is dependent on the structure of networks resembling Portland data, we firstly show how we model this resemblance. As pointed out in Section 4.1, the degree-distribution of locations (L) is well-approximated by a power law (see Definition 1.1.8) with exponent $\beta > 2$: i.e., the number of locations in L with degree i is close to $n_i = \frac{c|L|}{i^\beta}$, where c is a normalization constant. In the actual dataset, $\beta \sim 2.8$; we will work with an arbitrary constant $\beta > 2$. We let d_0 and d_1 denote the minimum and maximum location degrees, respectively; we will have $d_0 \ll d_1$, i.e., the location degrees exhibit large variations. On the other hand, the people degrees are small as expected, since a person cannot visit too many locations on a typical day; they are sharply concentrated around their average value, which will throughout be denoted w_p (and is approximately 4 in Portland data). Counting the number of edges from P and from L , we get

$$|P| \cdot w_p \sim \frac{\beta - 1}{\beta - 2} \cdot |L| \cdot d_0. \quad (5.1)$$

In urban settings, it is reasonable to assume that $|L|$ grows at a much slower rate than $|P|$ (after a city becomes larger than some critical size, one starts having lots of high-rise locations). Since w_p and $\beta > 2$ are small constants, (5.1) then implies that d_0 must be large. So we will assume for our graph models that $d_0 = \omega(1)$: i.e., a function of $|P|$ that increases unboundedly. (A typical choice could be some not-too-fast-growing function, such as $\text{polylog}(|P|)$.)

5.1 Chung-Lu's Model

We present the bipartite version of Chung-Lu's model [21] as follows.

Definition 5.1.1 (Chung-Lu's model) *Given two degree sequences, $D(P) = \{d(p) : p \in P\}$ for people set P and $D(L) = \{d(\ell) : \ell \in L\}$ for location set L , such that $\sigma = \sum_{p \in P} d(p) = \sum_{\ell \in L} d(\ell)$. Create a set of people P and a set locations L corresponding to the two degree sequences. Independently for each pair $p \in P$ and $\ell \in L$, put an edge between them with probability $\Pr[e(p, \ell)] = \frac{d(p)d(\ell)}{\sigma}$. Output the people and locations, and the randomly generated edges between them.*

5.1.1 FASTGREEDY in Chung-Lu's Model

We now prove that FASTGREEDY (Algorithm 1.1.5) is a $(1 + \epsilon)$ -approximation algorithm for networks resembling Portland data (see above) that are generated by Chung-Lu's model. For any $L' \subseteq L$, define $d(L') = \sum_{\ell \in L'} d(\ell)$. Consider first the case where $d(p)$ is the same for all $p \in P$ (recall that the people-degrees are highly concentrated around the mean). Before presenting the two key lemmas (Lemma 5.1.3 and Lemma 5.1.4), we show the following fact (Lemma 5.1.2) that will be used by them.

Lemma 5.1.2 *For any $x > 0$, we have $1 - x < e^{-x}$, and for all $0 < x \leq \frac{1}{2}$ and $\gamma \geq x$, we have $e^{-(1+\gamma)x} < 1 - x$.*

Proof It is easy to verify that for any $x > 0$, we have $1 - x < e^{-x}$. To prove the second inequality, we take the natural logarithm on both sides and obtain $\gamma > \frac{-\ln(1-x)}{x} - 1$. Since $-\ln(1-x) = \sum_{i=1}^{\infty} \frac{x^i}{i}$, we obtain $\gamma > \sum_{i=2}^{\infty} \frac{x^i}{i}$. On the other hand, we have $\sum_{i=2}^{\infty} \frac{x^i}{i} < \sum_{i=2}^{\infty} \frac{x^i}{2} = \frac{x}{2(1-x)}$, and $\frac{x}{2(1-x)} \leq x$ if $0 < x \leq \frac{1}{2}$. Therefore, when $0 < x \leq \frac{1}{2}$ and $\gamma \geq x$, we have $e^{-(1+\gamma)x} < 1 - x$. \square

The following lemma shows that with high probability, there is no subset of locations that can have a small sum of degrees and dominate a large fraction of people as well.

Lemma 5.1.3 *Let $0 < \epsilon_1 < 1$, $0 < \delta_1 < 1$, $\gamma \geq \frac{d_1}{|P|}$, where $d_1 \leq |P|/2$ (recall $\forall \ell \in L$, $d_0 \leq d(\ell) \leq d_1$). If $\frac{\epsilon_1 \delta_1^2}{2(1-\delta_1)} - \frac{|L|}{|P|} = \Omega(\frac{\ln |P|}{|P|})$, then with probability $1 - O(\frac{1}{|P|})$, there is no $L' \subset L$ such that $d(L') < |P| \cdot \frac{\ln \frac{1-\delta_1}{\epsilon_1}}{1+\gamma}$ and $|N(L')| \geq (1 - \epsilon_1)|P|$.*

Proof For any $p \in P$,

$$\Pr[p \notin N(L')] = \prod_{\ell \in L'} [1 - d(p)d(\ell)/\sigma],$$

where $\sigma = \sum_{p \in P} d(p) = \sum_{\ell \in L} d(\ell)$. By the assumption all people degrees are the same, hence $\sigma = |P| \cdot w_p$. Thus for any $p \in P$,

$$\begin{aligned} \Pr[p \notin N(L')] &= \prod_{\ell \in L'} (1 - d(\ell)/|P|) \\ &> \prod_{\ell \in L'} e^{-(1+\gamma)d(\ell)/|P|} \\ &= e^{-(1+\gamma)d(L')/|P|} \end{aligned}$$

where the first equality follows from the independence among the locations and the inequality follows from Lemma 5.1.2. Therefore, the expectation value of $|P \setminus N(L')|$ is at

least $|P| \cdot e^{-(1+\gamma)d(L')/|P|}$. It is easy to verify that when $d(L') < |P| \cdot \frac{\ln \frac{1-\delta_1}{\epsilon_1}}{1+\gamma}$, we have

$$\mathbf{E}[|P \setminus N(L')|] > |P| \cdot \frac{\epsilon_1}{1-\delta_1}.$$

By the Chernoff bound we obtain

$$\Pr[|N(L')| \geq (1-\epsilon_1)|P|] = \Pr[|P \setminus N(L')| < \epsilon_1|P|] < e^{-\frac{\epsilon_1 \delta_1^2}{2(1-\delta_1)}|P|}.$$

Since there are at most $2^{|L|}$ possible choices of L' , by the union bound we obtain the probability that there exists an $L' \subset L$ such that $d(L') < |P| \cdot \frac{\ln \frac{1-\delta_1}{\epsilon_1}}{1+\gamma}$ and $|N(L')| \geq (1-\epsilon_1)|P|$ is at most $e^{-|P|(\frac{\epsilon_1 \delta_1^2}{2(1-\delta_1)} - \frac{|L|}{|P|})}$. When $\frac{\epsilon_1 \delta_1^2}{2(1-\delta_1)} - \frac{|L|}{|P|} = \Omega(\frac{\ln |P|}{|P|})$, the probability is at most $O(\frac{1}{|P|})$. Therefore, with probability at least $1 - O(\frac{1}{|P|})$, there is no $L' \subset L$ such that $d(L') < |P| \cdot \frac{\ln \frac{1-\delta_1}{\epsilon_1}}{1+\gamma}$ and $|N(L')| \geq (1-\epsilon_1)|P|$. \square

Lemma 5.1.4 *Let $0 < \epsilon_2 < 1$, $0 < \delta_2 < 1$. If $\frac{(1-\delta_2-\epsilon_2)\delta_2^2}{2(1-\delta_2)} = \Omega(\frac{\ln |P|}{|P|})$, then with probability $1 - O(\frac{1}{|P|})$, for any $L' \subset L$ such that $d(L') \geq |P| \cdot \ln \frac{1-\delta_2}{\epsilon_2}$, we have $|N(L')| \geq (1-\delta_2-\epsilon_2)|P|$.*

Proof Similar to the proof of Lemma 5.1.3 and by Lemma 5.1.2, we have $\Pr[p \notin N(L')] < e^{-d(L')/|P|}$. Therefore

$$\mathbf{E}[|P \setminus N(L')|] < |P| \cdot e^{-d(L')/|P|}.$$

If $d(L') \geq |P| \cdot \ln \frac{1-\delta_2}{\epsilon_2}$ then $\mathbf{E}[|P \setminus N(L')|] < |P| \cdot \frac{\epsilon_2}{1-\delta_2}$ and $\mathbf{E}[|N(L')|] > (1 - \frac{\epsilon_2}{1-\delta_2})|P|$. By the Chernoff Inequality we obtain

$$\Pr[|N(L')| < (1-\delta_2-\epsilon_2)|P|] < e^{-\frac{(1-\delta_2-\epsilon_2)\delta_2^2}{2(1-\delta_2)}|P|}.$$

When $\frac{(1-\delta_2-\epsilon_2)\delta_2^2}{2(1-\delta_2)} = \Omega(\frac{\ln |P|}{|P|})$ the probability is at most $O(\frac{1}{|P|})$. Therefore, with probability at least $1 - O(\frac{1}{|P|})$, for any $L' \subset L$ such that $d(L') \geq |P| \cdot \ln \frac{1-\delta_2}{\epsilon_2}$, we have $|N(L')| \geq (1-\delta_2-\epsilon_2)|P|$. \square

Theorem 5.1.5 *For any constant $0 < \epsilon < 1$ such that $\epsilon = \Omega\left(\sqrt[3]{\frac{|L|}{|P|}}\right)$, with probability $1 - O(\frac{1}{|P|})$, the FASTGREEDY is a $\left((1 + \gamma) / \left(1 - \frac{1}{\log_2(2/\epsilon-1)}\right)\right)^{1+\frac{1}{\beta-2}}$ -approximation algorithm for the $(1 - \epsilon)$ -dominating set problem for Chung-Lu's model, where $\gamma \geq \frac{d_1}{|P|} = o(1)$.*

Proof Let $OPT \subset L$ be the minimum subset of locations that dominates a $(1 - \epsilon)$ -fraction of people. By Lemma 5.1.3 we obtain, with high probability (letting $\delta_1 = \epsilon/2$ and $\epsilon_1 = \epsilon$),

$$d(OPT) \geq |P| \cdot \frac{\ln \frac{2-\epsilon}{2\epsilon}}{1+\gamma}.$$

For any degree d , let $L_d = \{\ell \in L : d(\ell) \geq d\}$. Let d_2 be the smallest degree that $d(L_{d_2}) \leq |P| \cdot \frac{\ln \frac{2-\epsilon}{2\epsilon}}{1+\gamma}$, then $|OPT| \geq |L_{d_2}|$. Let d_3 be the largest degree that $d(L_{d_3}) \geq |P| \cdot \ln \frac{2-\epsilon}{\epsilon}$, then by Lemma 5.1.4 we obtain, with high probability (letting $\delta_2 = \epsilon_2 = \epsilon/2$), $|N(L_{d_3})| \geq (1 - \epsilon)|P|$. Let FG be the solution returned by the FASTGREEDY. The cost of FG is $|FG| \leq |L_{d_3}|$. Hence the approximation ratio is at most $\frac{|L_{d_3}|}{|L_{d_2}|}$. Without loss of generality, let $d(L_{d_2}) = |P| \cdot \frac{\ln \frac{2-\epsilon}{2\epsilon}}{1+\gamma}$ and $d(L_{d_3}) = |P| \cdot \ln \frac{2-\epsilon}{\epsilon}$. Due to the power-law property, we have $\frac{|L_{d_3}|}{|L_{d_2}|} \simeq \left(\frac{d(L_{d_3})}{d(L_{d_2})}\right)^{1+\frac{1}{\beta-2}} = \left((1 + \gamma) / \left(1 - \frac{1}{\log_2(2/\epsilon-1)}\right)\right)^{1+\frac{1}{\beta-2}}$. \square

In addition to comparing only the costs of FASTGREEDY and the optimum solution, we can also compare the amounts of people they dominate, and thus obtain a bicriteria result for the FASTGREEDY.

Theorem 5.1.6 *For any constant $0 < \epsilon < 1$ such that $\epsilon = \Omega\left(\sqrt[3]{\frac{|L|}{|P|}}\right)$, if the cost of the optimum solution OPT to the $(1 - \epsilon)$ -dominating set problem is $|OPT|$, then with high probability $(1 - O(\frac{1}{|P|}))$ the cost of the solution of FASTGREEDY to the $(1 - 2\epsilon)$ -dominating set problem is at most $(1 + \gamma)^{1+\frac{1}{\beta-2}} \cdot |OPT|$.*

Proof Let OPT be the optimum solution to the $(1 - \epsilon)$ -dominating set problem, by Lemmas 5.1.3 and 5.1.4 we obtain, with high probability $1 - O(\frac{1}{|P|})$ (letting $\delta_1 = \delta_2 =$

$$\epsilon_1 = \epsilon_2 = \epsilon = \Omega\left(\sqrt[3]{\frac{|L|}{|P|}}\right),$$

$$d(OPT) \geq |P| \cdot \frac{\ln \frac{1-\epsilon}{\epsilon}}{1+\gamma}.$$

On the other hand, let FG be the solution of FASTGREEDY to the $(1 - 2\epsilon)$ -dominating set problem, then with high probability $1 - O(\frac{1}{|P|})$,

$$d(FG) \leq |P| \cdot \ln \frac{1-\epsilon}{\epsilon}.$$

Similar to the proof of Theorem 5.1.5, $\frac{|FG|}{|OPT|} \simeq \left(\frac{d(FG)}{d(OPT)}\right)^{1+\frac{1}{\beta-2}} \leq (1+\gamma)^{1+\frac{1}{\beta-2}}.$ \square

In our case, $\frac{|L|}{|P|} \sim \frac{\beta-2}{\beta-1} \cdot \frac{w_p}{d_0}$ (see Equation 5.1), where $\beta > 2$, w_p is a small constant (in our experiments, $w_p \simeq 4$), $d_0 = \omega(1)$ (e.g. $\text{polylog}(n)$). We also have $d_1 = O(\sqrt{|P|}) < \frac{|P|}{2}$ which follows from the power-law property: $n_{d_1} \simeq (\beta-2)|P|w_p d_0^{\beta-2}/d_1^\beta \geq 1$ for $\beta > 2$, hence in Lemma 5.1.3 it is enough to set $\gamma = O(\frac{1}{\sqrt{|P|}})$. Using this, we show in Corollary 5.1.7 that the approximation ratios in both Theorems 5.1.5 and 5.1.6 are $(1+o(1))$. By a similar argument, we can prove that it is an $O(1)$ -approximation algorithm (with a small constant hidden in the $O(1)$) when the people-degrees vary somewhat.

Corollary 5.1.7 *Given $\beta > 2$ be a constant, if $\gamma = o(1)$ then the approximation ratio in Theorem 5.1.6 is $1 + o(1)$ with high probability. In addition, if $\frac{|P|}{|L|} = \omega(1)$ then the approximation ratio in Theorem 5.1.5 is also $1 + o(1)$ with high probability.*

Proof If $\beta > 2$ is a constant and $\gamma = o(1)$, it is easy to see that with high probability, the approximation ratio $(1+\gamma)^{1+\frac{1}{\beta-2}}$ is $1 + o(1)$. If we also have $\frac{|P|}{|L|} = \omega(1)$, then let $k = \Theta\left(\log \frac{|P|}{|L|}\right)$ and $\epsilon = 2^{-\Theta(k)}$, it is easy to verify that $\epsilon = \Omega\left(\sqrt[3]{\frac{|L|}{|P|}}\right)$ and $1 - \frac{1}{\log_2(2/\epsilon-1)} = 1 - 1/k$. Therefore with probability $1 - O(\frac{1}{|P|})$, the approximation ratio in Theorem 5.1.5 is $\left(\frac{1+\gamma}{1-1/k}\right)^{1+\frac{1}{\beta-2}} = 1 + o(1).$ \square

5.1.2 Domination by Sampling

FASTGREEDY and the traditional greedy algorithm require that the whole graph is accessible, in particular the degrees of locations. But collecting such data is a non-trivial task. In practice, any simulation (such as EpiSims) builds models based on a small sample of data from the whole population. This motivates the following question: *can one get a good solution to the domination problem by only sampling a fraction of the data?* We show here that this is possible, using the following SAMPLEDGREEDY algorithm.

Algorithm 5.1.8 (SampledGreedy) *Choose a set $P' \subset P$ by sampling each person independently with probability α . Compute the set $N(P')$ of locations visited by P' and run the FASTGREEDY on $P' \cup N(P')$, i.e., choose $L' \subset N(P')$ of highest degrees (restricted in P') such that $d_{P'}(L') \geq \alpha|P| \cdot \ln(1/\epsilon)$. L' is output as the solution for the whole graph.*

Theorem 5.1.9 *Assume $d_0 \geq c \cdot \log |P|$, where c is a certain constant. For any sampling probability $\alpha \geq 2 (\ln |P|)/d_0$, the solution L' chosen by the SAMPLEDGREEDY algorithm is also a $(1 + o(1))$ -approximation solution to the $(1 - \epsilon)$ -dominating set problem of the whole graph, with high probability.*

Proof Let $P' \subset P$ be the set of sampled people. Let $d_{P'}(\ell) = |N(\ell) \cap P'|$ be the restricted degree of ℓ in P' , we show below that $\forall \ell \in L, d_{P'}(\ell) = (1 + o(1))\alpha d(\ell)$, with high probability, given the assumption on α . As a result, if $d_{P'}(L') \geq (1 + o(1))\alpha|P| \ln(1/\epsilon)$, then $d(L') \geq |P| \ln(1/\epsilon)$, with high probability, and by Theorem 5.1.5, L' is a $(1 - \epsilon)$ -dominating set for P . The set L' is almost the same as the solution of running FASTGREEDY on the whole graph, because each location's new degree (restricted in P') is scaled by almost the same amount $(1 + o(1))\alpha$.

We now need to prove that $\forall \ell \in L, d_{P'}(\ell) = (1 + o(1))\alpha d(\ell)$ with high probability. This follows from a simple Chernoff bound. Firstly it is easy to verify that for each ℓ , $\mathbf{E}[d_{P'}(\ell)] = \alpha d(\ell)$. Because of independent sampling, and because $\alpha d(\ell) = \Omega(\ln |P|)$ for each ℓ , by Chernoff bound, the statement holds for any ℓ with probability at least $1 - 1/|P|^2$. The statement for all ℓ now holds by a union bound. \square

5.2 Configuration Model

The configuration model was first introduced by Bender and Canfield [13]. It was refined by Bollobás [15] and also Wormald [75] for studying random regular graphs. Structural results for general random graphs with arbitrary given degree sequence in the configuration model were first given by Molloy and Reed [56, 57]. We explore the bipartite version of configuration models with arbitrary given degree sequences and put it in a simple form in Figure 5.1.

Configuration Model

1. For each vertex $v \in L \cup P$, make $d(v)$ copies of v which are called stubs. Let S_L be the set of stubs of L and S_P the set of stubs of P . We have $|S_L| = |S_P| = \sigma$.
2. Fix an arbitrary order of stubs of L , e.g., $S_L(1), S_L(2), \dots, S_L(\sigma)$. Uniformly and randomly choose a permutation π of stubs of P in the following way:

For $i = 1$ to $\sigma - 1$ do

Uniformly and randomly choose an index j from $\{i, i + 1, \dots, \sigma\}$;
swap($S_P(i)$, $S_P(j)$).

The new sequence of stubs of P is $\pi(S_P)(1), \pi(S_P)(2), \dots, \pi(S_P)(\sigma)$.

3. For $1 \leq i \leq \sigma$, put an edge between $S_L(i)$ and $\pi(S_P)(i)$. For each vertex $v \in L \cup P$ merge all its stubs into the single vertex v . We obtain a multi-bipartite graph.

Figure 5.1: Description of the configuration model for generating random graphs.

Lemma 5.2.1 *The configuration model in Figure 5.1 generates all multi-bipartite graphs matching the two given degree sequences. Furthermore, each simple bipartite graph (if feasible) is generated with equal probability.*

Proof It is obvious that the configuration model generates all the multi-bipartite graphs matching the two given degree sequences. For any multi-bipartite graphs matching the degree sequences, let $k_{p,\ell} \geq 0$ be the number of edges between $p \in P$ and $\ell \in L$, obviously $\sum_{p \in P} k_{p,\ell} = d(\ell)$ and $\sum_{\ell \in L} k_{p,\ell} = d(p)$. There is a one-to-one mapping between the set of all multi-bipartite graphs matching the degree sequences and the collection of all feasible sets $\{k_{p,\ell} \geq 0 : \sum_{p \in P} k_{p,\ell} = d(\ell), \sum_{\ell \in L} k_{p,\ell} = d(p)\}$. Therefore the probability of generating a multi-bipartite graphs matching the degree sequences is equal to the probability of the occurrence of the corresponding set $\{k_{p,\ell} \geq 0 : \sum_{p \in P} k_{p,\ell} = d(\ell), \sum_{\ell \in L} k_{p,\ell} = d(p)\}$. This probability is equal to

$$\frac{1}{\sigma!} \cdot \prod_{\ell \in L} d(\ell)! \cdot \prod_{p \in P} \prod_{\substack{x_{p,\ell_{i+1}} = x_{p,\ell_i} - k_{p,\ell_i} \\ x_{p,\ell_0} = d(p)}} \binom{x_{p,\ell_i}}{k_{p,\ell_i}} = \frac{\prod_{\ell \in L} d(\ell)! \cdot \prod_{p \in P} d(p)!}{\sigma! \cdot \prod_{\ell \in L, p \in P} k_{p,\ell}}.$$

For simple bipartite graphs (if feasible), $k_{p,\ell} \in \{0, 1\}$ for all $p \in P$ and $\ell \in L$. Therefore, each simple bipartite graph is generated with equal probability $\frac{\prod_{\ell \in L} d(\ell)! \cdot \prod_{p \in P} d(p)!}{\sigma!}$. \square

Since there might be multi-edges generated by the configuration model, let $e^{(k)}(\ell, p)$ be the event that there are k edges between $\ell \in L$ and $p \in P$. Let $e^{(>k)}(\ell, p)$ be the event that there are more than k edges between $\ell \in L$ and $p \in P$. For a clean notation, let $e(\ell, p)$ denote $e^{(>0)}(\ell, p)$.

Lemma 5.2.2 *For any integer $a, b, c > 0$, let $g(a, b, c) = \prod_{0 \leq x \leq a} \left(1 - \frac{c}{b-x}\right)$, then*

$$g(a, b, c) = g(c, b, a) = \prod_{0 \leq x \leq c} \left(1 - \frac{a}{b-x}\right),$$

and

$$\max \left\{ \left(1 - \frac{c}{b-a}\right)^a, \left(1 - \frac{a}{b-c}\right)^c \right\} \leq g(a, b, c) = g(c, b, a) \leq \min \left\{ \left(1 - \frac{c}{b}\right)^a, \left(1 - \frac{a}{b}\right)^c \right\}.$$

Proof The first equality is easy to verify. The second inequality follows from the fact that $f(x) = \left(1 - \frac{c}{b-x}\right)$ is a decreasing function. \square

Lemma 5.2.3 For any $0 < x < 1$ and integer $k \geq 1$, $1 - kx \leq (1 - x)^k \leq 1 - kx + \frac{(kx)^2}{2}$.

Lemma 5.2.4 Let $d(v)$ be the degree of vertex $v \in L \cup P$. For any $\ell \in L$ and $p \in P$, and integer $0 \leq k \leq \min\{d(p), d(\ell)\}$, if $d(\ell) > 1$ and $d(p) > 1$, and $d(\ell) + d(p) \leq \sigma + k$, then the probability that there are exactly k edges between $\ell \in L$ and $p \in P$ is

$$\Pr[e^{(k)}(\ell, p)] = \frac{1}{k!} \cdot \prod_{0 \leq i < k} \frac{(d(\ell) - i) \cdot (d(p) - i)}{\sigma - i} \cdot \prod_{0 \leq j < d(p) - k} \left(1 - \frac{d(\ell) - k}{\sigma - k - j}\right)$$

and

$$\Pr[e^{(k+1)}(\ell, p)] = \frac{1}{k+1} \cdot \frac{(d(\ell) - k) \cdot (d(p) - k)}{\sigma - d(\ell) - d(p) + k + 1} \cdot \Pr[e^{(k)}(\ell, p)].$$

Proof If $d(\ell) + d(p) \leq \sigma + k$, then

$$\begin{aligned} \Pr[e^{(k)}(\ell, p)] &= \frac{\binom{d(\ell)}{k} \cdot \binom{d(p)}{k} \cdot k! \cdot \binom{\sigma - d(\ell)}{d(p) - k} \cdot (d(p) - k)! \cdot (\sigma - d(p))!}{\sigma!} \\ &= \frac{1}{k!} \cdot \prod_{0 \leq i < k} \frac{(d(\ell) - i)(d(p) - i)}{\sigma - i} \cdot \prod_{0 \leq j < d(p) - k} \left(1 - \frac{d(\ell) - k}{\sigma - k - j}\right). \end{aligned}$$

By comparing the terms of $\Pr[e^{(k)}(\ell, p)]$ and $\Pr[e^{(k+1)}(\ell, p)]$, we obtain the second equality. \square

Lemma 5.2.5 If $d(\ell) + d(p) > \sigma$ then $\Pr[e(\ell, p)] = 1$. If $d(\ell) + d(p) \leq \sigma$, then the probability of occurrences of edges between $\ell \in L$ and $p \in P$ is

$$\Pr[e(\ell, p)] = 1 - \prod_{0 \leq i < d(p)} \left(1 - \frac{d(\ell)}{\sigma - i}\right),$$

and

$$\frac{d(\ell)d(p)}{\sigma} \cdot \left(1 - \frac{d(\ell)d(p)}{2\sigma}\right) \leq \Pr[e(\ell, p)] \leq \frac{d(\ell)d(p)}{\sigma} \cdot \left(1 + \frac{d(p) - 1}{\sigma - d(p) + 1}\right).$$

When $\frac{d(\ell)d(p)}{\sigma} = o(1)$, we have $\Pr[e(\ell, p)] \approx \frac{d(\ell)d(p)}{\sigma}$.

Proof If $d(\ell) + d(p) > \sigma$, then no matter how to arrange the stubs, there is always at least one stub of p matching a stub of ℓ , and $\Pr[e(\ell, p)] = 1$.

If $d(\ell) + d(p) \leq \sigma$ then from Lemma 5.2.4 we obtain

$$\Pr[e(\ell, p)] = 1 - \Pr[e^{(0)}(\ell, p)] = 1 - \prod_{0 \leq i < d(p)} \left(1 - \frac{d(\ell)}{\sigma - i}\right).$$

By Lemma 5.2.2, we have $1 - \left(1 - \frac{d(\ell)}{\sigma}\right)^{d(p)} \leq \Pr[e(\ell, p)] \leq 1 - \left(1 - \frac{d(\ell)}{\sigma - d(p) + 1}\right)^{d(p)}$. By

Lemma 5.2.3, we have

$$1 - \left(1 - \frac{d(\ell)}{\sigma}\right)^{d(p)} \geq \frac{d(\ell)d(p)}{\sigma} - \frac{d(\ell)^2 d(p)^2}{2\sigma^2} = \frac{d(\ell)d(p)}{\sigma} \cdot \left(1 - \frac{d(\ell)d(p)}{2\sigma}\right)$$

and

$$1 - \left(1 - \frac{d(\ell)}{\sigma - d(p) + 1}\right)^{d(p)} \leq \frac{d(\ell)d(p)}{\sigma - d(p) + 1} = \frac{d(\ell)d(p)}{\sigma} \cdot \left(1 + \frac{d(p) - 1}{\sigma - d(p) + 1}\right).$$

Obviously when $\frac{d(p)}{\sigma} = o(1)$, we obtain from above that $\Pr[e(\ell, p)] \approx \frac{d(\ell)d(p)}{\sigma}$. \square

Proposition 5.2.6 *The expected number of multi-edges is*

$$\mathbf{E}[\#(\text{multi-edges})] = \sigma - \sum_{\ell \in L, p \in P} \left(1 - \prod_{0 \leq i < d(p)} \left(1 - \frac{d(\ell)}{\sigma - i}\right)\right) \leq \frac{\sum_{\ell \in L} d(\ell)^2 \cdot \sum_{p \in L} d(p)^2}{2\sigma^2}.$$

Proof Let $\mathbf{E}[\#e(\ell, p)]$ be the expected number of edges between $\ell \in L$ and $p \in P$. Since

$$d(p) = \sum_{\ell \in L} \mathbf{E}[\#e(\ell, p)] = \sum_{\ell \in L} \sum_{k=1}^{\min\{d(\ell), d(p)\}} k \cdot \Pr[e^{(k)}(\ell, p)]$$

we obtain

$$\begin{aligned}
\mathbf{E}[\#(\text{multi-edges})] &= \sum_{\ell \in L, p \in P} \sum_{k=2}^{\min\{d(\ell), d(p)\}} (k-1) \cdot \Pr[e^{(k)}(\ell, p)] \\
&= \sum_{\ell \in L, p \in P} \sum_{k=2}^{\min\{d(\ell), d(p)\}} k \cdot \Pr[e^{(k)}(\ell, p)] - \sum_{\ell \in L, p \in P} \sum_{k=2}^{\min\{d(\ell), d(p)\}} \Pr[e^{(k)}(\ell, p)] \\
&= \sum_{p \in P} \left(d(p) - \sum_{\ell \in L} \Pr[e^{(1)}(\ell, p)] \right) - \sum_{\ell \in L, p \in P} \left(1 - \Pr[e^{(0)}(\ell, p)] - \Pr[e^{(1)}(\ell, p)] \right) \\
&= \sigma - \sum_{\ell \in L, p \in P} \left(1 - \Pr[e^{(0)}(\ell, p)] \right) = \sigma - \sum_{\ell \in L, p \in P} \left(1 - \prod_{0 \leq i < d(p)} \left(1 - \frac{d(\ell)}{\sigma - i} \right) \right) \\
&\leq \sigma - \sum_{\ell \in L, p \in P} \frac{d(\ell)d(p)}{\sigma} \cdot \left(1 - \frac{d(\ell)d(p)}{2\sigma} \right) = \frac{\sum_{\ell \in L} d(\ell)^2 \cdot \sum_{p \in L} d(p)^2}{2\sigma^2}.
\end{aligned}$$

The last inequality follows from Lemma 5.2.5. \square

Corollary 5.2.7 *By our assumptions, i.e., all people have the same degree w_p , and the location degrees $d_0 \leq d(\ell) \leq d_1$ are power-law distributed with a constant exponent $2 < \beta < 3$, where $d_0 = \omega(1)$ and $d_1 = O(\sqrt{|P|})$. The expected fraction of multi-edges in all edges is $O\left(\frac{d_1}{|P|} \cdot \left(\frac{d_0}{d_1}\right)^{\beta-2}\right)$.*

Proof From Proposition 5.2.6 the expected fraction of multi-edges in all edges is at most $\frac{\sum_{\ell \in L} d(\ell)^2 \cdot \sum_{p \in L} d(p)^2}{2\sigma^3} = \frac{\sum_{\ell \in L} d(\ell)^2}{2\sigma \cdot |P|}$. Due to the power-law distribution, we have

$$\sum_{\ell \in L} d(\ell)^2 \approx \frac{\beta-2}{3-\beta} \cdot \frac{d_1^{3-\beta} - d_0^{3-\beta}}{d_0^{2-\beta} - d_1^{2-\beta}} \cdot \sigma = O\left(\left(\frac{d_0}{d_1}\right)^{\beta-2} \cdot d_1 \cdot \sigma\right).$$

Therefore the claim holds. \square

NOTE: In the Portland data, $\sigma = 6060679$, the estimation in Proposition 5.2.6 on the upper-bound of multi-edges is about 800. In the experiments, the actual number of multi-edges is between 600 and 700.

5.2.1 FASTGREEDY in the Configuration Model

By Lemma 5.2.5 the probability of occurrences of edges between $\ell \in L$ and $p \in P$ in the configuration model, is approximately $\frac{d(\ell)d(p)}{\sigma}$ (when $\frac{d(\ell)d(p)}{\sigma} = o(1)$), which is similar to the corresponding probability in Chung-Lu's model. The events of edges occurring among locations and people are negatively correlated, thus the extended Chernoff bounds [67] can be applied. This two observations hint the similarity of performance of the FASTGREEDY in the configuration model and Chung-Lu's model. In particular, we have the same two lemmas as in Section 5.1.1. In the following, we assume the uniform degree w_p for people as in Section 5.1.1. Since for any $L' \subset L$, if $d(L') > \sigma - w_p$ then obviously L' dominates all people in the configuration model. Hence we only consider those $L' \subset L$ such that $d(L') \leq \sigma - w_p$.

Theorem 5.2.8 *Lemmas 5.1.3 and 5.1.4 still hold for the configuration model. Hence Theorems 5.1.5 and 5.1.6 and Corollary 5.1.7 are also true for the configuration model. Furthermore, Theorem 5.1.9 also holds for the configuration model.*

Proof For any $p \in P$,

$$\begin{aligned} \Pr[p \notin N(L')] &= \prod_{0 \leq k \leq d(L')-1} \left(1 - \frac{w_p}{\sigma - k}\right) = \prod_{0 \leq k \leq w_p-1} \left(1 - \frac{d(L')}{\sigma - k}\right) \\ &> \left(1 - \frac{d(L')}{\sigma - w_p}\right)^{w_p} > e^{-(1+\gamma)d(L')/(|P|-1)} \end{aligned}$$

where the inequality follows from Lemma 5.1.2 and γ is specified in Lemma 5.1.3. On the other hand, we have

$$\begin{aligned} \Pr[p \notin N(L')] &= \prod_{0 \leq k \leq d(L')-1} \left(1 - \frac{w_p}{\sigma - k}\right) = \prod_{0 \leq k \leq w_p-1} \left(1 - \frac{d(L')}{\sigma - k}\right) \\ &< \left(1 - \frac{d(L')}{\sigma}\right)^{w_p} < e^{-d(L')/|P|}. \end{aligned}$$

Therefore, $|P| \cdot e^{-(1+\gamma)d(L')/(|P|-1)} < \mathbf{E}[|P \setminus N(L')|] < |P| \cdot e^{-d(L')/|P|}$. It is easy to verify that when $d(L') < (|P| - 1) \cdot \frac{\ln \frac{1-\delta_1}{\epsilon_1}}{1+\gamma} \approx |P| \cdot \frac{\ln \frac{1-\delta_1}{\epsilon_1}}{1+\gamma}$, $\mathbf{E}[|P \setminus N(L')|] > |P| \cdot \frac{\epsilon_1}{1-\delta_1}$. Meanwhile when $d(L') \geq |P| \cdot \ln \frac{1-\delta_2}{\epsilon_2}$, we have $\mathbf{E}[|N(L')|] \geq (1 - \frac{\epsilon_2}{1-\delta_2})|P|$. By the same argument in the proofs of Lemmas 5.1.3 and 5.1.4, substituting the extended Chernoff bound [67] for the Chernoff bound, we obtain the result. Due to these two lemmas, Theorems 5.1.5 and 5.1.6 and Corollary 5.1.7 for the FASTGREEDY are also true for the configuration model. The proof Theorem 5.1.9 is the same as that for Chung-Lu's model, substituting the extended Chernoff bound [67] for the Chernoff bound. \square

5.3 FASTGEN Model

Portland data is the single social network that we started with. In order to infer more complex structures from simple ones (e.g., degree sequences) and generalize from one network to a family of social networks that share basic structures, we have employed Chung-Lu's and configuration models to manipulate them by both theoretical (Sections 5.1, 5.2) and empirical (Section 5.4) means. Although similar in many features, in Section 5.5 we will see some fundamental difference between Chung-Lu's and configuration models, and why we need both of them. In terms of Chung-Lu's model, although its construction is related to the formation of some social networks, its running time of generating a single graph is $\Omega(|L| \cdot |P|)$ which is not efficient at all. In practice, we need to generate as many random graphs as possible to run simulations in order to discover the underlying consistent properties of certain structures or dynamics in the social networks. Although configuration model is efficient in terms of generating random graphs, the formation process of those graphs is not related to intrinsic dynamics of the formation of social networks. In order to stay close to the formation process of real social networks, yet generate random graphs

more efficiently than Chung-Lu’s model does, we designed a fast generation model FAST-GEN whose generating time is at most $O(\sigma \cdot \log |P|)$ and preserve most of Chung-Lu’s model’s properties, except the independence of creating edges. The edges in our model are negative correlated. Our model is a careful implementation of the approach of [68]. In realistic social networks, the negative correlation of creating edges in our model may be more a reasonable one than the independence in Chung-Lu’s model. Despite this difference, all the arguments we make in Chung-Lu’s model carry on to our fast generation model, via an extended Chernoff bound [67].

Unlike Chung-Lu’s and configuration models, where the two degree sequences are symmetric to the generation process of edges, our model distinguish these two degree sequences and therefore we have two version of the fast generation model, FASTGEN-1 and FASTGEN-2. Formally, we are given two disjoint sets P and L , and two positive integral degree sequences $D(P) = \{d(p) : p \in P\}$ and $D(L) = \{d(\ell) : \ell \in L\}$ such that $\sigma = \sum_{p \in P} d(p) = \sum_{\ell \in L} d(\ell)$ and $\max_{p \in P} d(p) \cdot \max_{\ell \in L} d(\ell) \leq \sigma$. Let the random variable $X_{p,\ell}$ denote the event that there is an edge between $p \in P$ and $\ell \in L$. FASTGEN-1 will be the following model, where the sequence of location-degrees equals $D(L)$ with probability 1. We will guarantee the following properties, by a careful implementation of the approach of [68]:

$$(a) \quad \forall p \in P, \forall \ell \in L, \Pr[X_{p,\ell} = 1] = \frac{d(p) \cdot d(\ell)}{\sigma};$$

$$(b) \quad \forall \ell \in L, \Pr[|\{p : X_{p,\ell} = 1\}| = d(\ell)] = 1;$$

$$(c) \quad \forall \ell \in L, \text{ the following “negative correlation” properties hold for all subsets } P' \subseteq P:$$

$$\Pr\left[\bigwedge_{p \in P'} (X_{p,\ell} = 0)\right] \leq \prod_{p \in P'} \Pr[X_{p,\ell} = 0],$$

$$\Pr\left[\bigwedge_{p \in P'} (X_{p,\ell} = 1)\right] \leq \prod_{p \in P'} [X_{p,\ell} = 1].$$

The first property implies that each edge is put in with the right probability. The second property ensures that the degrees of nodes in L are equal to the expected degrees. (Our algorithm can be trivially modified so that degrees in P are satisfied exactly, instead: that model is called FASTGEN-2.) The third property allows us to use Chernoff-like bounds [67], and we will use it later to show that several measures like overlap ratios, clustering coefficients etc. are preserved in this model. Meanwhile, due to these properties, the claims about the performance of FASTGREEDY for Chung-Lu's model in Section 5.1 also hold for FASTGEN model.

The discussion here is for model FASTGEN-1. Algorithm FASTGEN is a careful implementation of the approach of [68] and is described in Figure 5.2.

Lemma 5.3.1 *For any positive integer q and any index k , $\frac{\lfloor q/2 \rfloor}{q} < \sum_{i=0}^{q-1} \frac{w(\text{block}_{k+i})}{\sigma}$. $\frac{\max_{\ell \in L} d(\ell)}{q} \leq 1$. Also, the total number of blocks is at most $2(\max_{\ell \in L} d(\ell) + 1)$.*

Proof For any block_k and block_{k+1} , we have

$$\begin{aligned} & [w(\text{block}_k) + w(\text{block}_{k+1})] \cdot \max_{\ell \in L} d(\ell) \\ & > \sigma + [w(\text{block}_{k+1}) - d(p_{j_{k+1}})] \cdot \max_{\ell \in L} d(\ell) \\ & \geq \sigma. \end{aligned}$$

By grouping the blocks pairwise, we obtain $\lfloor \frac{q}{2} \rfloor \cdot \sigma < \sum_{i=0}^{q-1} w(\text{block}_{k+i}) \cdot \max_{\ell \in L} d(\ell)$. From inequality 5.2 we know $\sum_{i=0}^{q-1} w(\text{block}_{k+i}) \cdot \max_{\ell \in L} d(\ell) \leq q \cdot \sigma$. The first claim holds. Meanwhile we have $\sum_{i=1}^z w(\text{block}_i) = \sigma$, hence $\lfloor \frac{z}{2} \rfloor \cdot \sigma < \sigma \cdot \max_{\ell \in L} d(\ell)$. Therefore $z < 2(\max_{\ell \in L} d(\ell) + 1)$. \square

Algorithm FastGen-1

1. Order the elements in P in an arbitrary order, (p_1, p_2, \dots) .
2. Compute prefix sums $ps_i = \sum_{j=1}^i d(p_j)$ for $1 \leq i \leq |P|$, and define $ps_0 = 0$.
3. Partition the elements in P into a sequence of z blocks $block_1, block_2, \dots, block_z$ satisfying the following properties.
 - (a) Each block consists of a (contiguous) subsequence of P , e.g., $block_k = (p_{j_k}, p_{j_k+1}, \dots, p_{j_{k+1}-1})$.
 - (b) Let $w(block_k) = \sum_{p_j \in block_k} d(p_j)$.
 - (c) For each $k < z$, let p_{j_k} be the first element in $block_k$, the k -th block satisfies

$$\sigma - d(p_{j_{k+1}}) \cdot \max_{\ell \in L} d(\ell) < w(block_k) \cdot \max_{\ell \in L} d(\ell) \leq \sigma. \quad (5.2)$$

The last block satisfies $0 < w(block_z) \cdot \max_{\ell \in L} d(\ell) \leq \sigma$.

4. For each $\ell \in L$, choose $d(\ell)$ edges incident on ℓ by the following steps.
 - (a) Let $q_\ell = \lfloor \frac{\max_{\ell' \in L} d(\ell')}{d(\ell)} \rfloor \geq 1$. Group the blocks into super-blocks:

$$\begin{aligned} Block_1 &= (block_1, block_2, \dots, block_{q_\ell}), \\ Block_2 &= (block_{q_\ell+1}, block_{q_\ell+2}, \dots, block_{2 \cdot q_\ell}), \\ &\dots \end{aligned}$$

and so on. By Lemma 5.3.2, there are at most $2(d(\ell) + 1)$ super-blocks.

- (b) Let the random variable Y_i denote the event that there is an edge between ℓ and the super-block $Block_i$. Run the algorithm of [68] to determine if $Y_i = 1$ or 0.
- (c) Suppose $Block_i = (p_{t_i}, p_{t_i+1}, \dots, p_{t_i+j})$ which is a subsequence of the ordered people. If $Y_i = 1$, choose *one* $p \in Block_i$ with probability proportional to its weight $d(p)$ and add the edge (ℓ, p) . This can be done by first choosing an integer r uniformly at random from the interval $[ps_{t_i-1}, ps_{t_i+j})$, locating $p_x \in Block_i$ by running binary search on the (increasing) sequence $(ps_{t_i}, ps_{t_i+1}, \dots, ps_{t_i+j})$ and finding the index x such that $ps_{x-1} \leq r < ps_x$.

Figure 5.2: Description of the algorithm for fast generation of random graphs.

Lemma 5.3.2 *For each $\ell \in L$, the number of super blocks in step 4(a) of FASTGEN is at most $2(d(\ell) + 1)$.*

Proof The proof is similar to the proof of Lemma 5.3.1. \square

Lemma 5.3.3 *Algorithm FASTGEN satisfies the requirements of the fast generation model and runs in time $O(\sigma \cdot \log |P|)$.*

Proof From [68] we know that FASTGEN satisfies the requirements of the fast generation model. The prefix sums and blocks can be computed in time $|P| \leq \sigma$. For each $\ell \in L$, there are $O(d(\ell))$ super-blocks and hence the level-set algorithm of [68] and the binary search in each super-block runs in time $O(d(\ell) \cdot \log |P|)$. Therefore the total time is $\sum_{\ell \in L} O(d(\ell) \cdot \log |P|) = O(\sigma \cdot \log |P|)$. \square

5.3.1 Analytical Validation of FASTGEN-1

We now briefly explain why model FASTGEN-1 is a close approximation to Chung-Lu's model for the degree-distribution of the people-people graph, clustering coefficient, overlap ratios, etc. Similar remarks also hold for the distribution of shortest-path lengths.

Consider the degree-distribution of the people-people graph. Recall that a general model we are employing for the bipartite graph (P, L) is where: (a) the weight of any person is bounded by a constant B , and (b) the weight of the locations follow a power-law with exponent $\beta > 2$ with weights running in the range $[d_0, d_1]$, where d_0 is a slowly-growing function of $|P|$ such as $\text{polylog}(|P|)$. Since the number of locations with degree d_1 must be nonzero, it can be shown using the definition of power law that

$$d_1 \leq O(d_0 \cdot |L|^{1/\beta}) \leq O(|P|^{1/2 - \Omega(1)}).$$

Consider a person p . Its number of neighbors is distributed asymptotically as $\text{Poisson}(d(p))$, both in the model of [21] and in FASTGEN-1. Since $d(p) \leq B$, we have in particular that with high probability, the number of neighbors is at most $2 \ln |P| / (\ln \ln |P|)$. So, since each of these neighbors has degree at most $|P|^{1/2-\Omega(1)}$ as seen above, the total degree of these neighbors is at most $|P|^{1/2-\Omega(1)}$; thus, these neighboring locations of p will, in turn, have almost-disjoint neighborhoods in the set P , with high probability. Thus, the number of neighbors of p in the people-people graph is essentially the sum of the degrees of its neighbors, measured in the graph (P, L) . This sum is sharply concentrated around its mean, as it is a sum of independent random variables with mean $\omega(1)$ in the model of [21]; in FASTGEN-1, the concentration holds trivially with probability 1 (since the location-degrees equal the given degrees with probability 1).

A similar argument holds for the overlap ratios. Consider a subset $L' \subset L$. In Chung-Lu's model, the sum of degrees of locations in L' is sharply concentrated around its mean $d(L') = \sum_{\ell \in L'} d(\ell)$. In FASTGEN-1 this value exactly equals $d(L')$. In both models, the number of people adjacent to L' is also sharply concentrated around its mean:

$$\mathbf{E}[|N(L')|] = \sum_{p \in P} \left(1 - \prod_{\ell \in L'} \left(1 - \frac{d(p)d(\ell)}{\sigma} \right) \right).$$

The equality holds due to the independence of locations in both the Chung-Lu's model and FASTGEN-1. Therefore in terms of overlaps ratios, FASTGEN-1 closely approximates the Chung-Lu's model as well.

As for the clustering coefficient of the people-people graph, recall that it equals $(\sum_{p \in P} C(p)) / |P|$, where $C(p)$ is the number of edges $NE(p)$ in the neighborhood $N(p)$ of p in the people-people graph, divided by $\binom{N(p)}{2}$. Consider the random variable $C(p)$ for any fixed p , in both of our models of interest. As sketched above, the denominator $\binom{N(p)}{2}$ is highly concentrated around a value a ; thus, $\mathbf{E}[C(p)]$ is essentially $(1/a) \cdot \mathbf{E}[NE(p)]$.

Now, it can be shown that the main type of conditioning in the calculation of $\mathbf{E}[NE(p)]$, involves conditioning on a small constant number of edges incident on each of a set of locations L' . This conditioning poses no problems in the Chung-Lu's model, due to its full independence. It essentially poses no problem in model FASTGEN-1 either, since the locations make their edge-choices independently, and have expected degree $\omega(1)$.

It can also be shown that FASTGEN-2 is in general *not* a good approximation to [21] in the context of the above-seen parameters, basically because the weights $d(p)$ of the people are all small. In particular, conditional on the presence of an edge (p, l) in (P, L) , the distribution of other edges incident on p becomes altered significantly in FASTGEN-2. In addition, the outputs of FASTGEN-2 are not consistent with regards to different orderings of the input. Recall that in Step 1 of FASTGEN-2 (exchanging the roles of P and L in Figure 5.2), we need to order the elements in L . The following example shows that for the same degree sequences of L and P but different ordering of L , the outputs of FASTGEN-2 have different properties on the distributions of the degree sequence and clustering coefficients of the people-people graph.

Here is an example. Let all the people have the same input degree that equals 2, and let two locations have the same input degree that equals to $|P|/2$ (suppose $|P|$ is an even number), while all the other $|L| - 2$ locations have the same degree that equals the (small) value $\frac{|P|}{|L|-2}$. Thus the sum of degrees of all locations equals $2|P|$. Consider the following two orderings of the locations:

1. Divide the sequence of the locations into two parts. The first part consists of the two locations with degree $|P|/2$, the second part consists of all the locations with the small degree $\frac{|P|}{|L|-2}$;
2. Evenly divide the sequence of the locations into two parts, such that each part

contains half of the locations and the two locations with degree $|P|/2$ are in different parts.

Run FASTGEN-2 on the above two orderings separately. FASTGEN-2 groups locations into super blocks according to the ordering, and puts for each person exactly *two* edges between it and the locations. In both orderings, the locations are divided into two super blocks and each person has exactly one edge incident in each block. Suppose that the value $\frac{|P|}{|L|-2}$ is very small compared with $|P|$. It is easy to see that in the first run, almost all people will have degrees about $\frac{|P|}{2} + \frac{|P|}{|L|-2} - 2 \simeq \frac{|P|}{2}$ in the people-people graph, while in the second run, in the people-people graph (approximately) $1/2$ of people will have degrees about $\frac{|P|}{2} + \frac{|P|}{|L|-2} - 2 \simeq \frac{|P|}{2}$, and (approximately) $1/4$ of people will have large degrees close to n , and (approximately) $1/4$ of people will have small degrees about $\frac{2|P|}{|L|-2} - 2$. Similarly, in the first run, the clustering coefficients of almost all people are close to 1, while in the second run, (approximately) a *half* of the people will have clustering coefficients close to 1, and (approximately) a *quarter* of the people will have clustering coefficients about $1/2$, and the other *quarter* of the people will have relatively largely varied clustering coefficients due to their small average degrees which makes the concentration loose.

Although the ordering of P in FASTGEN-1 is also arbitrary, it basically behaves consistently for the parameters specified in our environment. For more comparisons of graphs generated by these two models, please see Section 5.4.

5.4 Empirical Comparison of Generated Graphs and Portland Data

We now present empirical results to strengthen the claims in Section 5.3. We also compare the experimental results of the random graph models with the original social contact

network for Portland. For this section we denote this network as the Portland data. Recall, that the degree sequence of this network is used as input by the random graph models. All the experiments were done on a shared machine, Sun UltraSPARC-III, 750 MHz CPU, 8.0 GB main memory. In all our experiments, we ignored isolated vertices generated by Chung-Lu’s model, i.e., vertices with degree 0. For example, in Table 5.1, the percentages of locations and people for the three models are the percentages of how many non-isolated locations and people are in the generated graph, compared to the Portland data. In all random graphs that we generate, there is a giant connected component consisting of almost all vertices. By projecting the people-location bipartite graph into the people-people graph and the location-location graph, we examine the sizes of the giant components in these two graphs separately. Since the Chung-Lu’s model is expensive to run, we only generated a few instances by this model. For all the random graph models we found that the values are sharply concentrated. Hence instead of taking averages, the data in Table 5.1 is for only one instance from each model. We study the following measures for these random graph models and the Portland data:

1. Degree distribution in the bipartite graph and in the induced people-people graph;
2. Overlap ratios;
3. Size of the giant component in the people-people and the location-location graph;
4. Shortest paths and clustering coefficients;
5. Quality of the FASTGREEDY algorithm for domination.

Table 5.1 compares some basic parameters between the Portland data and the graphs generated by Chung-Lu’s model and our FastGen models. The size of giant component

	Portland data	Chung-Lu	FastGen-1	FastGen-2
Num(locations)	181230	178746 (98.63%)	181230 (100%)	178668 (98.59%)
Sizeof(giant-locs.)	181192	178571	181088	178611
Num(people)	1615860	1507234 (93.28%)	1507291 (93.28%)	1615860 (100%)
Sizeof(giant-ppl.)	1615813	1507054	1507148	1615803
Num(edges)	6060679	6065637 (100.08%)	6060679 (100%)	6060679 (100%)
Avg-deg(ppl.)	3.7507	4.0227	4.0209	3.7507
Time(generating)		> 10 hours	< 40 seconds	< 30 seconds

Table 5.1: Comparing the basic structures for the Portland data and randomly generated graphs: number of locations, size of the giant connected component of locations, size of the giant connected component of people, number of edges in the bipartite graph, average degree of people in the bipartite graph, time of generating a graph. The percentages are the percents of the quantities of generated graphs compared to those of Portland data.

indicates the number of people (locations) in the giant connected component of the people-people (location-location) graph. Note that FASTGEN-1 preserves the number of locations and FASTGEN-2 preserves the number of people. Both of them preserves the number of edges. Please note that although the results of the configuration model are not presented in this table, they match those parameters of the Portland data very well, and it takes about 5 seconds to generate a random graph by the configuration model.

5.4.1 Degree Distributions

Figure 5.3 compares the degree distributions of *bipartite graphs* in the Portland data and the random graphs generated by Chung-Lu’s, FASTGEN-1, and FASTGEN-2 models. One can see from the figure that the three distributions are very close to each other. Note that FASTGEN-1 preserves location degrees, and FASTGEN-2 preserves people degrees. Also note that a large part of the degree distribution of locations exhibits a power-law. This part starts at degree approximately 20 and ends at degree about 200. For locations of degree k in $[20, 200]$, the number of them is proportional to $\frac{|L|}{k^\beta}$, where $\beta \simeq 2.8$, i.e., $|L_k| = \frac{c|L|}{k^{2.8}}$, where $L_k = \{l \in L : \deg(l) = k\}$, $20 \leq k \leq 200$, and $c \simeq 200$. The people

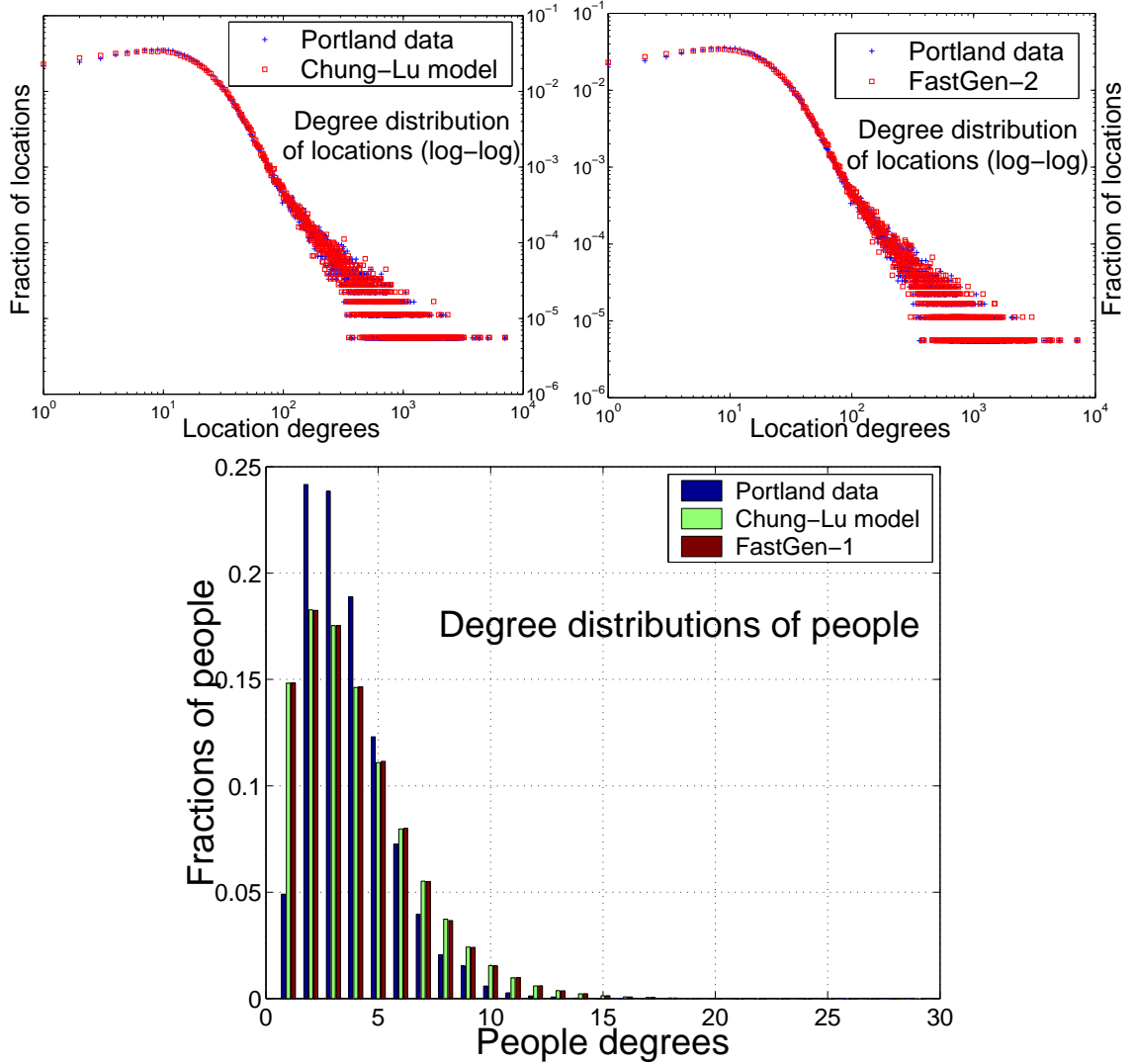


Figure 5.3: Comparison of (bipartite) degree distributions in the Portland data and graphs generated by Chung-Lu’s model, FastGen-1, and FastGen-2.

degrees resembles Poisson distributions with small means, and are upper-bounded by small constants.

Consider the degree distribution of the giant component of the *people-people graph*, G_P . The experimental results (Figure 5.4) and the theoretical analysis (Section 5.3) together confirm that the Chung-Lu’s and FASTGEN-1 models match the Portland data much more closely than FASTGEN-2.

5.4.2 Overlap Ratios and the FASTGREEDY Algorithm

We now study two variants of overlap ratios (Section 4.3) for locations in the bipartite graph. For any positive integer k , the point-overlap-ratio(k) is the overlap ratio for the set $L' = \{\ell \in L : d(\ell) = k\}$ and the cumulative-overlap-ratio(k) is the overlap ratio of the set $L' = \{\ell \in L : d(\ell) \geq k\}$. Note that all the overlap ratios are in $(0, 1]$. The higher the overlap ratio is, the better the FASTGREEDY algorithm performs for the dominating set problem. The plots of the overlap ratios are shown in Figure 5.5. One can see from the figure that corresponding to the degree distribution of locations, there is a large part of the overlap ratios exhibiting the power-law property.

From experiments (Figure 5.5 and the theoretical analysis (Section 5.3) we can conclude that the overlap ratios of graphs generated by FASTGEN-1 are close to the graphs generated by Chung-Lu’s model and both of them are close to the Portland data. On the other hand, the overlap ratios of graphs generated by FASTGEN-2 are much higher than the corresponding overlap ratios in graphs generated by the other two models and Portland data. Thus we can expect that FASTGREEDY should perform much better for

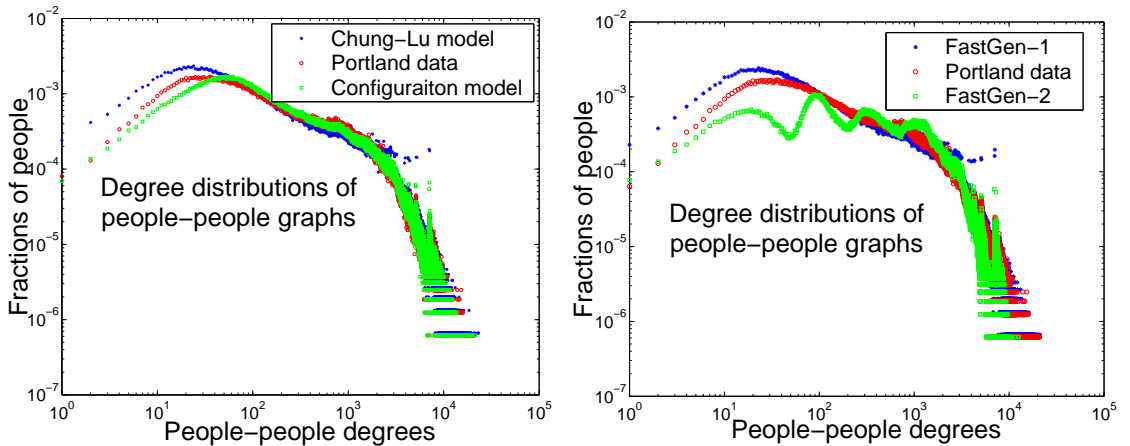


Figure 5.4: Comparison of degree distributions of people-people graphs of Chung-Lu’s model, Portland data, the configuration model, FASTGEN-1, and FASTGEN-2.

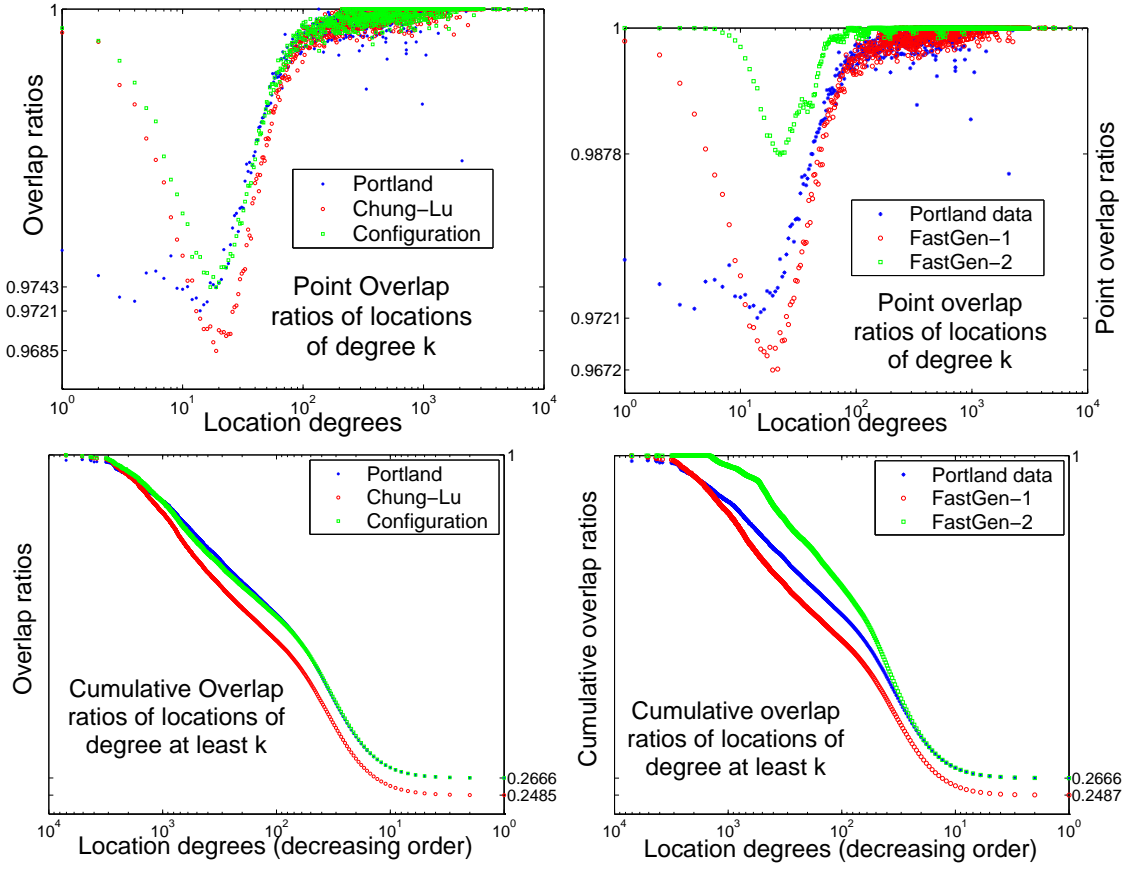


Figure 5.5: Overlap ratios of locations in the Portland data and the four models.

the graphs generated by FASTGEN-2 than by the others. This is supported by the plots of the performance of FASTGREEDY in Figure 5.6.

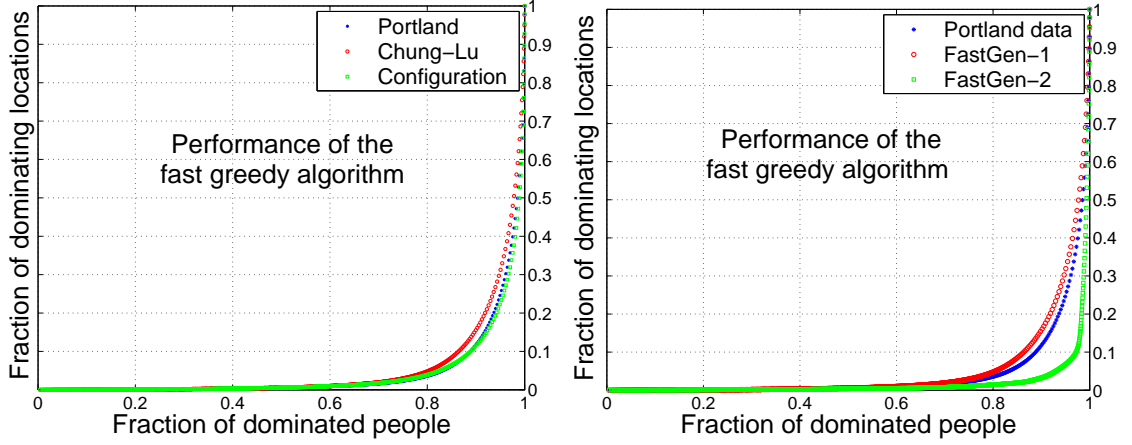


Figure 5.6: Performance of FASTGREEDY for the dominating set problems.

We also compare our FASTGREEDY (4.3.1) with the traditional greedy algorithm (1.1.5) that has the $O(\log n)$ approximation ratio. To make the comparison fair, we use the Frugal-FASTGREEDY algorithm (4.3.2). In this comparison, we still call this modified algorithm FASTGREEDY. The comparison between the FastGreedy and the regular greedy is in Table 5.2.

	Portland data	Chung-Lu	FASTGEN-1	FASTGEN-2
FASTGREEDY	47.35%	58.37%	59.60%	27.90%
GREEDY	41.23%	54.47%	55.78%	27.54%
Time(FASTGREEDY)	< 15 seconds	< 15 seconds	< 15 seconds	< 15 seconds
Time(GREEDY)	> 4 hours	> 5 hours	> 7 hours	> 2 hours

Table 5.2: Performance of FASTGREEDY and GREEDY (see 1.1.5) for the Portland data and graphs generated by Chung-Lu’s model and the FASTGEN models. Percentages denote the sizes of the dominating sets (compared to the whole set of locations). Seconds and hours denote the time needed to run these algorithms.

5.4.3 Shortest Paths and Clustering Coefficients

Two popular measures people study on social networks are the shortest path distribution and the clustering coefficients. Often, these give important connectivity information for the contact networks. Both of these can be computed easily in polynomial time, but usual algorithms take $\Omega(|P|^2)$ time, which is infeasible for such large graphs. This motivates faster methods to approximate these distributions. We show simple random sampling based algorithms for these two problems, and show their empirical performance. For practical reasons, we examine the shortest paths and the clustering coefficients only in the giant component of the people-people graph.

Since $\log_2 |P| \simeq 20$, we did seven experiments each by uniformly and independently sampling about a hundred vertices (i.e., people) in the Portland data and computed the shortest-path spanning tree for each sampled vertex. The fraction of shortest paths of distance i in the giant-component is estimated by the fraction of paths of distance i in all the shortest-path spanning trees. Similarly we did the same experiments for the Chung-Lu’s model, FASTGEN-1 and FASTGEN-2. For each model, nine experiments were done and in each experiment about a hundred vertices were sampled. The means and the standard deviations of the experiments for the Portland data and the three models are presented in Table 5.3. From the table we can see that the distance between most pairs of people in the giant connected component is 2 or 3.

We also estimate the average clustering coefficient (cc) (see Definition 1.1.9) of the giant component of the people-people graph by sampling about a hundred vertices in each experiment. We did seven experiments respectively for the Portland data and the graphs generated by the three models. We calculated the mean and the standard deviation of the seven experiments. These values are presented in Table 5.4.

		len=1	len=2	len=3	len=4	len=5	len=6
Portland data	mean	8.0806×10^{-4}	0.2666	0.7126	0.0200	4.8687×10^{-5}	5.6341×10^{-9}
	std.	9.1226×10^{-5}	0.0149	0.0138	0.0049	8.8549×10^{-5}	9.8875×10^{-9}
Chung-Lu	mean	9.7835×10^{-4}	0.4264	0.5665	0.0061	1.0460×10^{-6}	
	std.	1.1151×10^{-4}	0.0241	0.0223	0.0025	5.1511×10^{-7}	
FASTGEN-1	mean	0.0010	0.4161	0.5756	0.0072	4.6721×10^{-7}	
	std.	1.5816×10^{-4}	0.0300	0.0282	0.0028	2.0310×10^{-7}	
FASTGEN-2	mean	8.4945×10^{-4}	0.3622	0.6344	0.0026	2.9173×10^{-7}	
	std.	6.9460×10^{-5}	0.0177	0.0181	0.0019	5.3671×10^{-7}	

Table 5.3: Means and standard deviations of fractions of shortest paths of different lengths in the single-source shortest-path spanning trees sampled from the giant component of the people-people graphs.

In addition to the average value calculated from clustering coefficients of all vertices, we also computed the global clustering coefficients (see Definition 1.1.10) of Portland data and graphs generated by Chung-Lu’s model and the configuration model. Table 5.5 compares these values. They match very well.

5.5 A Generic Framework

The random process of putting edges in bipartite graphs is independent in Chung-Lu’s model, whereas it is negatively correlated in the configuration model. In either case, a

	Portland data	Chung-Lu	FASTGEN-1	FASTGEN-2
cc mean	0.6376	0.6161	0.6235	0.7021
cc std.	0.0167	0.0315	0.0236	0.0201

Table 5.4: Means and standard deviations of clustering coefficients of sampled vertices in the giant components of people-people graphs. Lower bounds and approximated lower bounds to the clustering coefficients of giant components of people-people graphs.

	Edges	Triangles	Len.-2 paths	Clust. coef.
Portland	1077247259	631174770092	3349781340086	0.565268
Configuration	1080704055	629038069416	3308711641941	0.570347
Chung-Lu	1087038876	636683653469	3731989178759	0.511805

Table 5.5: Numbers of edges, triangles, and length-2 paths, and clustering coefficients in the people-people graph (1615860 people) in the Portland data and the Chung-Lu’s model.

Chernoff-like bound [19, 45, 67] can be applied easily to them. In Chung-Lu’s model the probability of the occurrence of edges between a person p and a location ℓ is exactly proportional to the products of their given degrees, whereas it is approximately that value in the configuration model. In terms of these two properties, Chung-Lu’s model and the configuration model are similar and one should expect similar properties in these two models. On the other hand, the configuration model generates a graph in linear time $O(\sigma)$ and is much more efficient than Chung-Lu’s model (in time $\Omega(|P| \cdot |L|)$) and even our FASTGEN model (in time $O(\sigma \cdot \log |P|)$). However in realistic social networks, there usually exists an underlying probability distribution of associating a person and a location, with the constraint of the resource limit which can be characterized as their degrees. Social networks are formed by realizations of random processes obeying these probability distributions and the constraint (the degrees). Chung-Lu’s model is such an example that assumes the simple probability that is proportional to the product of two degrees. By substituting another probability distribution for the one in Chung-Lu’s model, and still matching the degree sequence, one can obtain a new random graph model. The configuration model however, is not flexible to adapt to other probability distributions given the degree sequence, though it approximates the special probability distribution (as in Chung-Lu’s model) very well.

A model for generating random bipartite graphs of any feasible degree sequences and probability distributions was given in [37], where the events of occurrences of edges are negatively correlated. We can view this model as a generalization of Chung-Lu’s model and our FASTGEN model. Although being a general model matching both the degree sequence and the probability distribution of making edges, the model in [37] is inefficient and runs in time $O(|V| \cdot |E|)$, where $|V|$ is the number of vertices and $|E|$ is the number

of edges in the bipartite graph.

An interesting observation of social networks may help us bypass the bottleneck of generating time of certain models without sacrificing their intrinsic merits. Many social networks exhibits prominent *community* structures, where the whole networks can be decomposed into smaller sub-networks, called *communities*. Each community may play a special social role among the whole social networks. Each edge of the social network can represent a relationship between two people in the social network and each person can play several roles by having different relationships with others. Therefore, an *edge-partition* of the social network partitions the roles of different communities, whereas a person can belong to several communities to play multi-roles in the social network. Besides the natural semantic meaning, the community structure can also help us generate random graph efficiently. For example, suppose the graph $G(V, E)$ can be edge-partitioned into n subgraphs $\bigcup_{i=1}^n G_i(V_i, E_i)$, where $V = \bigcup_i V_i$, $E = \bigcup_i E_i$ and $E_i \cap E_j = \emptyset$ for any $i \neq j$. Independently generate each community G_i using an appropriate model, say the one in [37]. Since the communities are an edge-partition of the whole graph, they are independent of each other in terms of the random generation process, for that the process is focused on putting edges between pair of vertices. Further assume that $|E_i| \leq k$ for some $k > 0$ and all $1 \leq i \leq n$, which means that each community can only sustain activities up to a threshold value k . The total running time of using [37] is

$$\sum_{i=1}^n O(|V_i| \cdot |E_i|) \leq k \cdot O\left(\sum_{i=1}^n |V_i|\right).$$

If $\sum_{i=1}^n |V_i| = O(d \cdot |V|)$ and $k \cdot d \ll |E|$, then generating subgraphs one by one is much faster than generating the graph as a whole. Also, due to the edge-partition, the communities are independently of each other and can be generated in parallel, which improves the generating time significantly.

On the other hand, if the communities are not given and parallel platforms are not available, we can run the edge-partitioning algorithm in [38, 42] which is defined in 5.5.1.

Problem 5.5.1 (Edge partitioning problem) *Given a graph $G = (V, E)$ and a positive integer k , an edge partition of graph G is a collection of subgraphs $\{G_i(V_i, E_i) \subseteq G(V, E)\}$ induced by the partition $E = \bigcup_i^n E_i$, $E_i \cap E_j = \emptyset$ for any $i \neq j$, such that $|E_i| \leq k$ for all i 's. The problem is to find an edge partition that minimize $\sum_{i=1}^n |V_i|$, where n is the number of parts.*

This problem is NP-hard and was addressed by Goldschmidt, Hochbaum, Levin, and Olinick [42] and by us [38]. [42] gave an $O(\sqrt{k})$ -approximation algorithm to the edge partitioning problem and we [38] gave an $O(|V|^{1/3})$ -approximation algorithm. These two results are not comparable in general.

Based on the above discussion, we present a framework for generating random social networks. We suppose that natures of each community have different requirement on their generating process. For example, for communities is formed by intrinsic probability distributions of connecting pairs of people, using Chung-Lu's model (or our FASTGEN model) or the model in [37] is a good choice. If the intrinsic random process of creating edges is not important, then the configuration model is a good choice for the sake of generating times. When using models of large time complexities (e.g., Chung-Lu's model and the model in [37]), one may want to apply edge partitioning algorithms [38, 42] to reduce the time complexity. If the community structure is given, one can apply different models to different communities for the best match of their social functionals.

Chapter 6

Algorithms for Quarantining and Vaccination Problems

A significant part of epidemiology deals with understanding vaccination and quarantining policies, and answering questions such as whom to vaccinate. The potential for answering such questions is greatly reduced if we stick to uniform mixing models. In this section we show how three policy planning problems can be formulated in terms of the network structure, and, in some cases, can be solved efficiently.

6.1 Modeling the Efficacy of Vaccination Policies

The problem is to formulate a simple model for determining the efficiency of a vaccination scheme. Using the network structure, one way to model this problem is to consider the shattering problem defined in section 4.2 – the vaccination is effective if the giant component size goes down. In this model, the discussion in Section 4.2 shows that the policy of vaccinating all people of degree larger than some threshold is not very effective. Another model could be to look at the diameter of the giant component, or the average distance between a pair of nodes.

In the first model above, the best vaccination strategy would be to choose a set of nodes to vaccinate (and remove them from the network), so that each component in the remaining graph becomes small. If we are given a parameter $\rho \in (0, 1)$, and require each component to have at most ρ fraction of the nodes, then this reduces to the ρ -separator problem [52]. This problem is NP-hard, and [52] gives a polylogarithmic approximation to the optimum. Unfortunately, this algorithm is unlikely to scale for such large graphs,

and an interesting problem is to find a faster approximation algorithm.

6.2 The Quarantining and Vaccination Problems

A basic problem for public health workers that we consider next is: given a set $I \subset P$ of infected nodes in the graph G_P , what is the optimal set of nodes to vaccinate or quarantine? There are two aspects of cost here: one is the number (or total cost) of people to vaccinate or quarantine, and the other is the set of people who are reachable from I after deleting the nodes that get vaccinated (this models a highly infectious disease). We describe two formulations of this problem here.

(a) The VACCINATION PROBLEM (VP): given a graph $G(V, E)$, an initial infected set I , a parameter C , and cost $c(v)$ for each node v , choose a set of nodes $S \subseteq V \setminus I$ having cost $c(S) = \sum_{v \in S} c(v) \leq C$, such that the size of the set

$$A_G(I, S) = \{v \mid v \text{ has a path from some } w \in I \text{ in } G(V \setminus S)\}$$

is minimized. The set S is the people who are vaccinated, and the set $A_G(I, S)$ models the set of people that could get infected by a highly infectious disease.

(b) The QUARANTINING PROBLEM (QP): given a graph $G(V, E)$, an initial infected set I , a parameter B , and cost $c(e)$ for each edge e , choose a cut (S, \bar{S}) such that $I \subseteq S$, $|S| \leq B$, and the cost of the cut is minimized. Unlike vaccination, quarantining involves cutting down some of the contacts - this is captured by the edge deletions in this model.

We show that the VP and QP problems are NP-complete and give a bicriteria approximation for them, using network flows.

Lemma 6.2.1 *The VP and QP problems are NP-hard.*

Proof We will show that the decision versions of these problems, with a bound C on

the (edge/vertex) cut size and bound B on the number of nodes that get infected, are NP-complete. These decision problems are clearly in NP. The hardness is by reduction from the NP-complete *bipartition problem* [39].

An instance of bipartition involves a graph $G(V, E)$, with cost $c(e)$ on each edge e , and bound C ; the problem is to decide whether there is a partition (V_1, V_2) such that $|V_1| = \lfloor n/2 \rfloor, |V_2| = \lceil n/2 \rceil$ and $c(V_1, V_2) \leq C$, where $c(V_1, V_2) = \sum_{v_1 \in V_1, v_2 \in V_2} c(v_1, v_2)$. We will always assume w.l.o.g. that $C \leq \sum_{e \in E} c(e)$. We first describe the reduction from bipartition to QP. Given an instance $G(V, E)$ of Bipartition, we construct a new graph G' in the following manner. Add a new vertex s to G , with an edge from s to each vertex in V . Each edge $(s, v), v \in V$ has cost $c(s, v) = \alpha = \sum_{e \in E(G)} c(e) + 1$. Now we consider the QP problem on graph G' with $I = \{s\}$ (the infected set), with bounds $C' = C + \lceil n/2 \rceil \alpha$ and $B' = \lfloor n/2 \rfloor + 1$. We show that this instance is feasible if and only if the given bipartition instance is. Suppose there is a solution (S, \bar{S}) to the QP problem with bounds C' and B' with $s \in S$. We first claim that $|S| = B'$. For, if $|S| < B'$, the cost of the cut (S, \bar{S}) would be at least

$$(n + 1 - |S|)\alpha \geq (n + 1 - \lfloor n/2 \rfloor)\alpha = \lceil n/2 \rceil \alpha + \alpha > \lceil n/2 \rceil \alpha + C = C'.$$

On the other hand, given that $|S| = B'$ and the cost of the cut (S, \bar{S}) is at most C' , we claim that the *bipartition* $(V_1 = S \setminus \{s\}, V_2 = V \setminus S)$ is a solution with cost at most C to the bipartition problem on $G(V, E)$. This is because in the QP problem, the cost of the cut (S, \bar{S}) contributed by $\{s\}$ is exactly $\lceil n/2 \rceil \alpha$, thus the cost of the cut contributed by $V_1 = S \setminus \{s\}$ is at most $C' - \lceil n/2 \rceil \alpha = C$. Conversely, if (V_1, V_2) is a solution to the bipartition problem such that $|V_1| = \lfloor n/2 \rfloor$, it is easy to verify that $(S = V_1 \cup \{s\}, V_2)$ is a solution to the QP problem; thus QP is NP-complete.

For the VP problem, the reduction needs to be modified, since we are dealing with

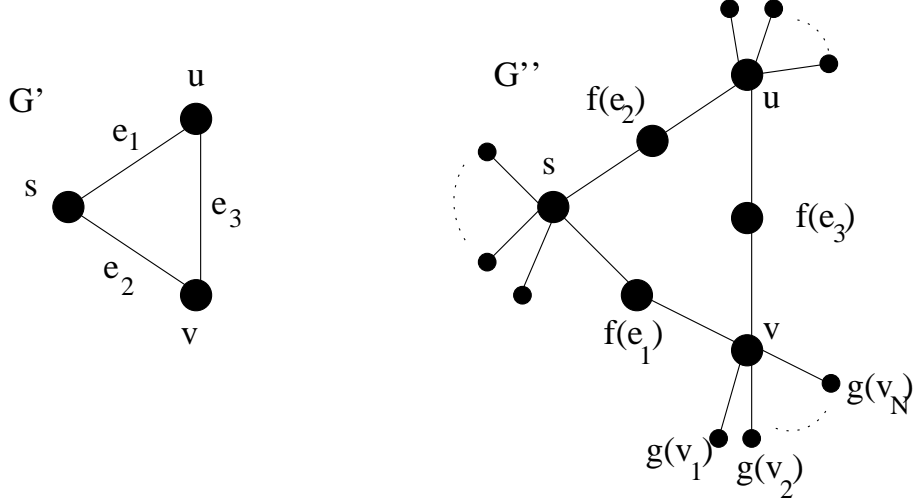


Figure 6.1: The transformations done in the reduction for VP.

vertex cuts. From the graph G' constructed above, we construct another graph G'' in the following manner. $V(G'') = V_A \cup V_B \cup V_C$, where $V_A = V(G')$, $V_B = \{f(e) : e \in E(G')\}$, and $V_C = \{g(v, 1), \dots, g(v, N) : v \in V_A\}$, for $N = n^4$ being a large number; Figure 6.1 illustrates the reduction on a small example. Here, f and g are just one-to-one indexing functions. V_B is partitioned into two sets $V_B = V_{B_1} \cup V_{B_2}$, where $V_{B_1} = \{f(e) : e = (s, v) \in E(G')\}$ and $V_{B_2} = \{f(e) : e = (v, w) \in E(G'), v, w \neq s\}$. The vertex costs are defined in the following manner: (i) for each $v = f(e) \in V_{B_1}$, $c(v) = \alpha = \sum_{e' \in E(G')} c(e') + 1$; (ii) for each $v = f(e) \in V_{B_2}$, $c(v) = c(e)$; and (iii) letting M be the sum of the costs of all vertices in V_B , we define, for each $v \in (V_A \cup V_C)$, $c(v) = M$. The edges $E(G'')$ are defined in the following manner: (i) for each edge $e = (s, v) \in E(G')$, we have the edges $(s, f(e))$ and $(f(e), v)$ that go between V_A and V_B ; (ii) for each edge $e = (v, w) \in E(G'), v, w \neq s$, we have the edges $(v, f(e))$ and $(f(e), w)$ that go between V_A and V_B ; and (iii) we have the edges $(v, g(v, i))$ for each $v \in V_A, i = 1, \dots, N$ (these edges go between V_A and V_C). We now consider the VP problem on G'' with $I = \{s\}$ and bounds $C' = C + \lceil n/2 \rceil \alpha$ and $B' = (N + 2)\lceil n/2 \rceil + n^2$. We argue below that the bipartition instance has a solution if

and only if the VP instance on G'' has a solution with the given B', C' .

Suppose the above VP problem is feasible, and $S \subseteq V(G'')$ is a solution, with $c(S) \leq C', |A_{G''}(I, S)| \leq B'$. Then, we argue now that $(V_1 = A_{G''}(I, S) \cap V_A, V_2 = V \setminus V_1)$ is a solution to the bipartition problem with cost C . Since each vertex $w \in V_A \cup V_C$ has cost $c(w) = M > C'$, $S \subseteq V_B$. Also, since for each vertex $v \in V_{B_1}$, $c(v) = \alpha > \sum_{v' \in V_{B_2}} c(v')$, it follows that $|V_{B_1} \cap S| \leq \lceil n/2 \rceil$. Every vertex in V_{B_1} must lie either in $A_{G''}(I, S)$ or in S , and, therefore, $|V_{B_1} \cap A_{G''}(I, S)| \geq \lfloor n/2 \rfloor$; this, in turn, implies that $|V_A \cap A_{G''}(I, S)| \geq \lfloor n/2 \rfloor$. Also, for each $v \in A_{G''}(I, S) \cap V_A$, we have for all i that $g(v, i) \in A_{G''}(I, S)$, because $S \cap V_C = \emptyset$; therefore, the bound $|A_{G''}(I, S)| \leq B'$ shows that $|V_A \cap A_{G''}(I, S)| \leq \lfloor n/2 \rfloor$, for that $N = n^4 \gg n^2$. The above statements together imply $|V_1 = A_{G''}(I, S) \cap V_A| = \lfloor n/2 \rfloor$, and this further implies $|V_{B_1} \cap S| = \lceil n/2 \rceil$. Since $c(S) = \sum_{v \in S \cap V_{B_1}} c(v) + \sum_{v \in S \cap V_{B_2}} c(v)$, it follows that $\sum_{v \in S \cap V_{B_2}} c(v) \leq C$, and, therefore, the cut (V_1, V_2) has cost at most C in G . Conversely, suppose (V_1, V_2) is a solution to the bipartition problem of cost C . Then, it can be verified easily that $S = \{f(e) : e \in \text{cut}(V_1, V_2)\} \cup \{f(e) : e = (s, v) \in E(G'), v \in V_2\}$ is a solution to the VP problem: $c(S) = C + |V_2|\alpha = C'$, and $|A_{G''}(I, S)| = (|V_1 \cup \{s\}|)(N + 2) + |\{e \in V_{B_2} : e \in V_1 \times V_1\}| \leq B'$. Together, these imply that the VP problem is also NP-complete. \square

To simplify the discussion, we first describe an approximation for QP, and later describe how this can be modified to work for the VP problem. In our discussion, we will assume that the graph is directed - an undirected graph can be made directed by putting edges in both directions.

6.3 Bicriteria Results for QP

We describe a $(1 + 2\epsilon, 1 + 2/\epsilon)$ approximation algorithm for QP, for any given $\epsilon \in (0, 1)$. That is, if the least number of newly-infected people in an instance of QP with bound C is denoted by OPT , our approximation algorithm produces a solution with the cost of the cut at most $(1 + 2\epsilon)C$ and at most $(1 + 2/\epsilon)OPT$ newly-infected people.

Let $G(V, E)$ be an instance of QP, with $I \subseteq V$ being the infected set, and C being the bound on the cost of the cut. The algorithm is outlined in Figure 6.2.

Input: Graph G , cost $c(e)$ on each edge e , the infected set I and $\epsilon > 0$.
 For $i = -\log_{(1+\epsilon^2)}(n/(C\epsilon)), \dots, -1, 0, 1, \dots, \log_{(1+\epsilon^2)}(C\epsilon)$ repeat the following steps and choose the best solution:

1. Let $\beta \leftarrow (1 + \epsilon^2)^i$;
2. Construct a new graph G' : Add a source s and sink t to G . Add edges $(s, v), \forall v \in I$ with cost ∞ and edges $(v, t), \forall v \notin I$ with cost β . All edges of G retain their old costs;
3. Compute the minimum s-t cut (S, \bar{S}) , where $s \in S$. The candidate solution is $(S \setminus \{s\}, \bar{S} \setminus \{t\})$.

Figure 6.2: A Bicriteria Algorithm for QP

Lemma 6.3.1 *For any given $\epsilon > 0$, the algorithm in Figure 6.2 produces a solution with cut cost at most $(1 + 2\epsilon)C$ and at most $(1 + 2/\epsilon)OPT$ infected nodes, where OPT is the number of infected people in the optimal solution with cut cost C .*

Proof We will first show that if $\beta = \epsilon C / OPT$, the solution obtained by solving the minimum cut is an $(1 + \epsilon, 1 + 1/\epsilon)$ approximation. Let (X, \bar{X}) be the optimal solution with $c(X, \bar{X}) \leq C$, $I \subseteq X$ and $|X \setminus I| = OPT$. Then the solution $(X \cup \{s\}, \bar{X} \cup \{t\})$ has cut cost of $C + \beta \cdot OPT$. Now suppose the minimum cut in G' is $(S \cup \{s\}, \bar{S} \cup \{t\})$, for $S \subseteq V(G)$. Clearly, $I \subseteq S$ (else the edges of cost ∞ have to cross the cut). Let C' be the

cost of the cut (S, \bar{S}) in G . Then, $C' + (|S| - |I|)\beta \leq C + \beta \cdot OPT$. Setting $\beta = \epsilon C / OPT$, this inequality implies that $C' \leq (1 + \epsilon)C$ and $|S| - |I| \leq (1 + 1/\epsilon)OPT$.

Since we do not know the value of OPT , we cannot try out this exact value of β . By trying out all the powers of ϵ , given that $1 \leq OPT \leq n$, we can approximate this ratio. Suppose, the best solution is obtained for $\beta \in [\beta_0/(1 + \delta), \beta_0(1 + \delta)]$, where $\beta_0 = \epsilon C / OPT$ and $\delta = \epsilon^2$, and let this solution be the cut (S, \bar{S}) in G . As above, we have $c(S, \bar{S}) + (|S| - |I|)\beta \leq C + \beta OPT$. Since $\beta \leq \beta_0(1 + \delta)$, we have $c(S, \bar{S}) \leq C(1 + \epsilon(1 + \delta)) \leq C(1 + 2\epsilon)$, since $\epsilon < 1$. Similarly, $|S| - |I| \leq OPT(1 + 2/\epsilon)$. \square

6.4 A Bicriteria approximation for VP.

Figure 6.3 shows the bicriteria algorithm for VP, and is a modification of the algorithm in Figure 6.2. This algorithm assumes that the graph is directed; if G is undirected, we just put in edges directed both ways before calling it. Lemma 6.4.1 shows its correctness.

Input: Graph G , cost $c(v)$ on each vertex v , the infected set I and $\epsilon > 0$.
For $i = -\log_{(1+\epsilon^2)}(n/(C\epsilon)), \dots, -1, 0, 1, \dots, \log_{(1+\epsilon^2)}(C\epsilon)$ repeat the following steps and choose the best solution:

1. Let $\beta \leftarrow (1 + \epsilon^2)^i$;
2. Construct a new edge weighted graph G' : For each vertex v , we split it into two vertices v_{in} and v_{out} , and add the edge $e = (v_{in}, v_{out})$ having cost $c(e) = c(v)$ for $v \in V \setminus I$, and $c(e) = \infty$ for $v \in I$. In addition, for each edge (v, w) in the original graph, we add the edges $e = (v_{out}, w_{in})$ having cost $c(e) = \infty$.
3. Add a source s and sink t to G' . For each vertex $v \in I$, add edges $e = (s, v_{in})$ with $c(e) = \infty$. For each vertex $v \in V \setminus I$, add edges $e = (v_{out}, t)$ with $c(e) = \beta$.
4. Compute the minimum s-t cut (X, \bar{X}) , where $s \in X$. The candidate solution is $S = \{v : (v_{in}, v_{out}) \text{ lies in the cut}\}$, and $A(I, S) = \{v : v_{out} \in X\}$.

Figure 6.3: A Bicriteria Algorithm for VP

Lemma 6.4.1 *For any given $\epsilon > 0$, the algorithm in Figure 6.3 produces a solution with*

cut cost (i.e., the number of people vaccinated) at most $(1+2\epsilon)C$ and at most $(1+2/\epsilon)OPT$ infected nodes, where OPT is the number of infected people in the optimal solution with C vaccinations.

Proof We firstly argue that the algorithm produces a feasible solution to the VP problem. Note that there is a finite $s - t$ cut - deleting all edges of the form (v_{in}, v_{out}) , where $v \in V \setminus I$, has cost $c(S) = \sum_{v \in V \setminus I} c(v)$, and $|A(I, S)| = 0$. This corresponds to the trivial solution of vaccinating all people in $V \setminus I$, and thus no one will be infected. Another trivial solution of vaccinating no people and thus all will be infected corresponds to a finite $s - t$ cut of deleting all edges of the form (v_{out}, t) , where $v \in V \setminus I$. In this finite $s - t$ cut solution, $c(S) = \beta|V \setminus I|$ and $S = \emptyset$. The two trivial finite cuts are not necessary the minimum cuts, but their existence implies that our argument will not be vacuous.

Next, note that any edge of the form $e = (s, v_{in})$, or $e = (v_{in}, v_{out})$ for $v \in I$, or $e = (v_{out}, w_{in})$ for some v, w has $c(e) = \infty$; therefore, no such edge can be part of the minimum cut (X, \bar{X}) - the only edges that can be in the cut are of the form (v_{in}, v_{out}) , or (v_{out}, t) for $v \in V \setminus I$. Also, there can be no vertex v such that $v_{out} \in X, v_{in} \in \bar{X}$ - this would require some edge of cost ∞ to be in the cut. This implies that the set S separates the set $A(I, S)$ from $V \setminus (S \cup A(I, S))$ in the original graph G . Also, $c(X, \bar{X}) = c(S) + \beta|A(I, S)|$, where $c(S) = \sum_{v \in S} c(v)$.

As in the proof of Lemma 6.3.1, we will first show that if $\beta = \epsilon C / OPT$, the solution is an $(1 + \epsilon, 1 + 1/\epsilon)$ approximation. Let S' be the optimal solution to VP , with $c(S') \leq C$, and $|A(I, S')| = OPT$. Since (X, \bar{X}) is the minimum $s - t$ cut, it must be the case that $c(S) + \beta|A(I, S)| \leq c(S') + \beta OPT$. Setting $\beta = \epsilon C / OPT$, this inequality implies that $C' \leq (1 + \epsilon)C$ and $|A(I, S)| \leq (1 + 1/\epsilon)OPT$.

The rest of the argument, to take care of the fact that the “optimal” value of β is not known, is the same as in the proof of Lemma 6.3.1. \square

Chapter 7

Simulation Based Dynamic Analysis

We now turn to simulation based analysis of the disease dynamics. Our results show that graph theoretic analysis can provide useful insights and guide the simulation based dynamic analysis. Much of the discussion in this section is from our article [29], which focused on smallpox. As noted below, many quantitative results such as total number of cases depend on details of the disease model.

Our results on high expansion suggest that the disease is likely to spread quickly if it is not controlled at an early stage (see Chapter 4). However, exactly how the number of casualties depends on response delay and what constitutes *early enough* depend on disease-specific factors such as incubation period and probability of transmission, as well as scenario-specific factors such as the means of introduction. Because these dependencies cannot be easily determined from analysis of the static social network, we turn to simulation. See the supplemental information of [29] for details about the particular disease model used in the simulation experiments. At present, there is no consensus on models of smallpox. The model used in our study captures many features on which there is widespread agreement and allows us to vary poorly understood properties through reasonable ranges [29]. Our model includes the following features (see [29] for definitions of some of the terms below):

- The incubation period is a truncated Gaussian distribution;
- The prodromal period is 3-5 days;
- The infectious period is 4 days, during which infectivity decreases exponentially;

- Death occurs 10-16 days after the rash develops in 30% of normal cases. 95% of susceptibles exposed for three hours to a person at minimum infectivity for three hours will become infected. The remaining 5% have extremely high or low susceptibilities, mimicking some anecdotal transmission incidents;
- Vaccination is assumed to be 100% effective pre-exposure, and its effectiveness is less if it is administered some time after the exposure.

The model also includes hemorrhagic variants with a shorter incubation period that are ten times as infectious and invariably fatal. Importantly, EpiSims does not specify a value for R_0 (see Section 2.1 for definition), the basic reproductive number. This parameter reflects how many people in a susceptible population are directly infected by the introduction of a single infective individual. R_0 is a convolution of transmission rates and contact patterns, and EpiSims carries out the convolution for us. The implied value of R_0 is the ratio of numbers of people in the first and original cohorts; these estimates obviously include the effects of the simulated response strategy. For the set of experiments reported below, R_0 ranges from 0.4 to 3.4. In these scenarios, aerosolized smallpox was distributed indoors at busy locations over several hours, infecting on the order of 1000 people. We assumed that the presence of smallpox was detected on the tenth day after the attack. We studied the sensitivity of the number of casualties to three factors: *mitigation efforts*, *delay in implementing mitigation efforts*, and *whether people move about while infectious*. We simulated a passive (do nothing) *baseline* and three active responses:

- Mass vaccination covering 100% of the population in four days (“mass”);
- Targeted vaccination and quarantine with unlimited resources (“targeted”);
- The same targeted response, using only half as many contact tracers and vaccinators

(“limited”).

The choice of individuals for targeted vaccination and quarantining is based on our graph theoretic analysis and uses two heuristic methods: (i) individuals in the first neighborhood of the infected people are candidates for vaccination or quarantine based on high expansion of the network suggesting early action, and (ii) individuals with higher degrees (contacts) are more likely to be chosen from this subset. A more detailed experimental design that uses cultural measures and sophisticated graph theoretic measures is currently being conducted.

For a movie showing the spatial spread of disease under two different response strategies, see [29]. Figure 7.1 compares the efficacy of these strategies. For each strategy, we plot (on a logarithmic scale) the ratio of the cumulative number of deaths by day 100 to the number initially infected. The absolute numbers are less important than the rank and relative sizes of gaps between the points. Also shown are the effects of 4, 7, or 10 day delays in implementing the response. For each of the responses including the baseline, we allowed infected people to isolate themselves by withdrawing to the home. This could be due to either the natural history of the disease, which incapacitates its victims, or actions taken by public health officials encouraging people to stay home. The results are grouped according to time of withdrawal to the home:

1. EARLY: people withdraw before they become infectious, producing the lowest estimates for R_0 ;
2. LATE: people withdraw roughly 24 hours after they have become infectious; and
3. NEVER people carry on their daily activities unless they die. The extreme cases are unrealistic, but are shown here because they demonstrate the existence of a clear

transition.

The study shows that time of withdrawal to the home is by far the most important factor, followed by delay in response. This indicates that targeted vaccination is feasible when combined with fast detection. Ironically, the actual strategy used is much less important than either of these factors. Overall, these results suggest a much greater efficacy for targeted strategies than suggested by the results of Kaplan, Craft, and Wein [46]. It is not clear what accounts for the difference. Possibilities include: differences in mixing rates, differences in the distribution of incubation periods, and differences in transmissivity.

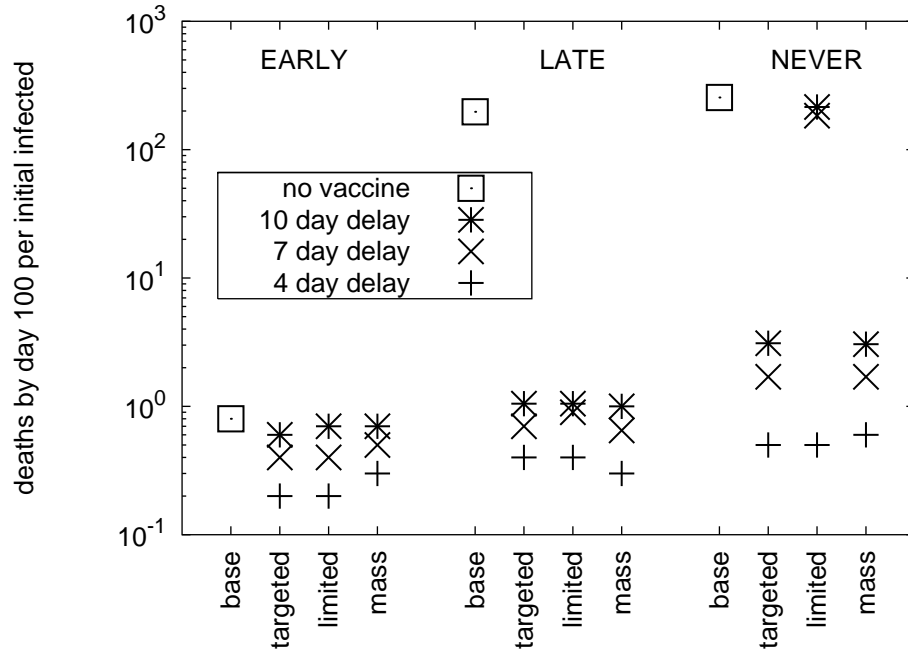


Figure 7.1: Cumulative number of deaths per number of initial infected, in case of a smallpox outbreak in downtown Portland, under a number of different response strategies.

Chapter 8

Conclusions and Future Work

In this thesis we studied the structure of a large urban social network, Portland data (Chapter 4), and models that can generate random graphs resembling this social network (Chapter 5). These structures include not only certain basic ones, e.g., degree distributions, clustering coefficients, and shortest paths, but also some new ones, like expansions and overlap ratios, which proved to be very important in studying social networks in a large urban area. We also showed the temporal and demographic structures of Portland data. We examined the targeted vaccination strategy via a shattering process (Section 4.2) and showed that due to the high expansion, this strategy is not efficient. On the other hand, after observing high overlap ratios in Portland data, we designed an efficient disease-detection algorithm (modeled through the dominating set problem, Section 4.3).

All these structures and performance of algorithms were rigorously analyzed in two random graph models, Chung-Lu’s model and configuration model (Chapter 5). We also designed a fast-generation model (Section 5.3) that inherits all the properties of Chung-Lu’s model except the independence among edges. But our model captures another important feature of social networks that Chung-Lu’s model doesn’t, i.e., negative correlation which corresponds to resource constraints in social environments. We also proposed a generic framework (Section 5.5) for generating random social networks matching given degrees and required probability distributions. In order to make this framework efficient, we present a community method and an edge partitioning method to decompose the whole graph into smaller subgraphs and generate them independently.

Besides the FASTGREEDY algorithm for disease early-detection, we designed two efficient approximation algorithms for vaccination and quarantining problems on social contact networks (Chapter 6).

We also presented two simulation tools, TRANSIMS and EpiSims, developed by Los Alamos National Laboratory (Chapter 3). The study shows (Chapter 7) that time of withdrawal to the home is by far the most important factor, followed by delay in response. This indicates that targeted vaccination (Section 4.2) is feasible when combined with fast detection (Section 4.3).

8.1 Future Work

We are aware that there are still lots of important problems to solve for social networks, from computing basic structures to designing efficient disease defending strategies, to modeling them through random graphs, and to large-scale simulations, etc. We list some of them in the following:

- **Basic structures**

How to count the number of triangles in a graph efficiently? How to compute the clustering coefficients and shortest path distributions more efficient through sampling methods?

- **Efficient strategies**

How do we design efficient vaccination and quarantining strategies? How do we combine early detection and targeted vaccination strategies?

- **Graph models**

How to efficiently generate random graphs matching multiple properties? E.g.,

matching both degree sequences and clustering coefficients. How to efficiently generate random graphs matching degree sequences and arbitrary probability distributions? How to generate random social networks not only satisfying specified topological structures, but also matching specific semantic features, e.g., demographic structure, temporal structures, etc.?

- **Simulations**

How do we accurately simulate real disease spreading, e.g., SARS and bird influenza?

How do we use simulation to assist decision makers?

BIBLIOGRAPHY

- [1] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
- [2] R. Albert, A. Barabási, and H. Jeong. Diameter of the world wide web. *Nature*, 401:103–131, 1999.
- [3] R. Albert, A. Jeong, and A. L. Barabási. Attack and error tolerance of complex networks. *Nature*, 406:378–382, 2000.
- [4] R. Albert, H. Jeong, and A.-L. Barabási. Attack and error tolerance of complex networks. *Nature*, 406:378–382, 2000.
- [5] N. Alon and J. Spencer. *The probabilistic method*. Wiley-Interscience, second edition, 2000.
- [6] R. M. Anderson and R. M. May. *Infectious Diseases of Humans*. Oxford University Press, Oxford, 1991.
- [7] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases and Its Applications*. Hafner Press, New York, 1975.
- [8] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
- [9] C. Barrett, K. Bisset, R. Jacob, G. Konjevod, and M. Marathe. An experimental analysis of a routing algorithm for realistic transportation networks. In *Proceedings of European Symposium on Algorithms*, 2002.

- [10] C. Barrett and *et al.* TRANSIMS (TRansportation ANalysis SIMulation System). Technical Report LA-UR-99-1658, LA-UR-99-2574–LA-UR-99-2580, Los Alamos National Laboratory, 1999.
- [11] C. Barrett, R. Jacob, and M. Marathe. Formal language constrained path problems. *SIAM Journal of Computing*, 30(3):809–837, 2001.
- [12] C. Barrett, J. P. Smith, and S. Eubank. Modern epidemiology modeling. Technical Report LA-UR-04-4176, Los Alamos National Laboratory, 2004.
- [13] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory A*, 24:196–307, 1978.
- [14] J. Berry, L. Fleischer, W. E. Hart, and C. Phillips. Sensor placement in municipal water networks. In *Proceedings of World Water and Environmental Resources Conference*, 2003.
- [15] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal on Combinatorics*, 1:311–316, 1980.
- [16] B. Bollobás and O. Riordan. Mathematical results on scale free graphs. In S. Bornholdt and H. Schuster, editors, *Handbook of graphs and networks*. Wiley-VCH, Berlin, November 2002.
- [17] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proceedings of the 9th World Wide Web Conference*, pages 309–320, 2000.

- [18] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letter*, 85:5468–5471, 2000.
- [19] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.
- [20] F. Chung and L. Lu. The average distance in random graphs with given expected degrees. *Proceedings of National Academy of Science (USA)*, 99:15879–15882, 2002.
- [21] F. Chung and L. Lu. Connected components in random graphs with given degree sequences. *Annals of Combinatorics*, 6:125–145, 2002.
- [22] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin. Resilience of the internet to random breakdowns. *Physical Review Letter*, 85:4626–4628, 2000.
- [23] C. Cooper and A. Frieze. A general model of web graphs. *Random Structures and Algorithms*, 22:311–335, 2003.
- [24] K. Dietz. Transmission and control of arbovirus diseases. In K. L. Cooke, editor, *Epidemiology*, pages 104–121. SIAM, Philadelphia, 1975.
- [25] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, 2001.
- [26] P. Domingos and M. Richardson. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, 2002.

- [27] K. Elveback, J. Fox, and E. Ackerman. An influenza simulation model for immunization studies. *American Journal of Epidemiology*, 103:152–165, 1976.
- [28] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.
- [29] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180–184, 2004. Supplemental information: <http://www.nature.com/nature/journal/v429/n6988/extref/nature02541-s1.htm>.
- [30] S. Eubank, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, and N. Wang. Structural and algorithmic aspects of massive social networks. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 2004.
- [31] S. Eubank, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, and N. Wang. Structure of social contact networks and their impact on epidemics. In J. Abello and G. Cormode, editors, *AMS-DIMACS Special Volume on Epidemiology*. American Mathematical Society, 2005.
- [32] S. Eubank and J. Smith. Scalable, efficient epidemiological simulation. In *Proceedings of Symposium on Applied Computing*, 2002.
- [33] S. Eubank and *et al.* Episims assessment of responses to smallpox attack (report to the office of homeland security). Technical Report LA-CP-02-254, Los Alamos National Laboratory, 2002.
- [34] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On the power law relationships of the internet topology. *Computer Communication Review*, 29(4):251–262, 1999.

- [35] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
- [36] A. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Structures and Algorithms*, 10:5–42, 1997.
- [37] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding in bipartite graphs. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, 2002.
- [38] R. Gandhi, S. Khuller, A. Srinivasan, and N. Wang. Approximation algorithms for channel allocation problems in broadcast networks. In *Proceedings of International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 2003.
- [39] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
- [40] E. Garfield. It’s a small world after all. *Current Contents*, 43:5–10, 1979.
- [41] O. Goldreich and D. Ron. On testing expansion in bounded degree graphs. Technical Report TR00-020, Electronic Colloquium on Computational Complexity, 2000.
- [42] O. Goldschmidt, D. Hochbaum, A. Levin, and E. Olinick. The sonet edge-partition problem. *Networks*, 41:13–23, 2003.
- [43] J. Guare. *Six Degrees of Separation: A Play*. Vintage, New York, 1990.
- [44] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42:599–653, 2000.

- [45] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.
- [46] E. H. Kaplan, D. L. Craft, and L. M. Wein. Emergency response to a smallpox attack: the case for mass vaccination. *Proceedings of the National Academy Sciences (U.S.A.)*, 99:10935–10940, 2002.
- [47] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining*, 2003.
- [48] J. M. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.
- [49] J. M. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 163–170, 2000.
- [50] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 57–65, 2000.
- [51] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. The web as a graph. In *Proceedings of the 19th ACM Symposium on Principles of Database Systems*, pages 1–10, 2000.
- [52] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for multicommodity flow problems with applications to approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

- [53] I. Longini, E. Halloran, A. Nizam, and Y. Yang. Containing pandemic influenza with antiviral agents. *American Journal of Epidemiology*, 159(1):623–633, 2004.
- [54] L. Meyers, M. E. J. Newman, M. Martin, and S. Schrag. Applying network theory to epidemics: Control measures for outbreaks of mycoplasma pneumonia. *Emerging Infectious Diseases*, 9:204–210, 2003.
- [55] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
- [56] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–179, 1995.
- [57] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability and Computing*, 7:295–305, 1998.
- [58] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.
- [59] M. E. J. Newman. Random graphs as models of networks. In S. Bornholdt and H. G. Schuster, editors, *Handbook of Graphs and Networks*. Wiley-VCH, Berlin, 2003.
- [60] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–25, 2003.
- [61] M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66:035101, 2002.
- [62] M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Physical Review E*, 68:036122, 2003.

- [63] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118, 2001.
- [64] J. Park and M. E. J. Newman. The origin of degree correlations in the internet and other networks. *Physical Review E*, 68, 2003.
- [65] R. Patel, I. Longini, and E. Halloran. Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. Technical Report 04-07, Department of Biostatistics, Rollins School of Public Health, Emory University, Atlanta, 2004.
- [66] S. V. Pemmaraju. Equitable colorings extend Chernoff-Hoeffding bounds. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, pages 924–925, 2001.
- [67] J. P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM Journal on Discrete Mathematics*, 8:223–250, 1995.
- [68] A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 588–597, 2001.
- [69] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.
- [70] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.
- [71] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
- [72] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

- [73] D. J. Watts. Networks, dynamics, and the small world phenomenon. *American Journal of Sociology*, 105:493–592, 1999.
- [74] D. J. Watts. *Small Worlds*. Princeton University Press, Princeton, 1999.
- [75] N. C. Wormald. The asymptotic connectivity of labelled regular graphs. *Journal of Combinatorial Theory B*, 31:156–167, 1981.
- [76] E. Zwingle. Cities. *National Geographic Magazine*, 202:70–99, 2002.